Facial Modeling and Animation

Jörg Haber[1], Demetri Terzopoulos[2], Nadia Magnenat-Thalmann[3], Volker Blanz[4]

[1] MPI Informatik, Saarbrücken, Germany, haberj@mpi-sb.mpg.de
[2] Media Research Lab, New York University, USA, dt@cs.nyu.edu
[3] MIRALab, University of Geneva, Switzerland, thalmann@miralab.unige.ch
[4] MPI Informatik, Saarbrücken, Germany, blanz@mpi-sb.mpg.de

Abstract

In this tutorial we present an overview of the concepts and current techniques that have been developed to model and animate human faces. We introduce the research area of facial modeling and animation by its history and applications. As a necessary prerequisite for facial modeling, data acquisition is discussed in detail. We describe basic concepts of facial animation and present different approaches including parametric models, performance-, physics-, and image-based methods. State-of-the-art techniques such as MPEG-4 facial animation parameters, mass-spring networks for skin models, and face space representations are part of these approaches. We furthermore discuss texturing of head models and rendering of skin and hair, addressing problems related to texture synthesis, bump mapping with graphics hardware, and dynamics of hair. Typical applications for facial modeling and animation such as speech synchronization, head morphing, and forensic applications are presented and explained.

1. Outline

start	topic	presenter(s)
8:30	Outline of the tutorial	
8:35	History & application areas of FAM	D. Terzopoulos
9:00	Anatomy of the human head	J. Haber
9:15	Data acquisition	V. Blanz
9:45	Overview: FAM techniques	J. Haber
10:00	*coffee break*	
10:30	Parametric models	N. Magnenat-Thalmann
11:00	Performance-driven animation	D. Terzopoulos
11:20	Physics-based approaches	D. Terzopoulos
12:00	*lunch break*	
14:00	Image-based systems	V. Blanz
14:40	Forensic applications	J. Haber
15:05	Speech synchronization	N. Magnenat-Thalmann
15:30	*coffee break*	
16:00	Texturing faces	J. Haber
16:20	Rendering: skin, wrinkles, hair	N. Magnenat-Thalmann + J. Haber
16:50	Morphing & caricatures	V. Blanz
17:15	Questions & discussion	all

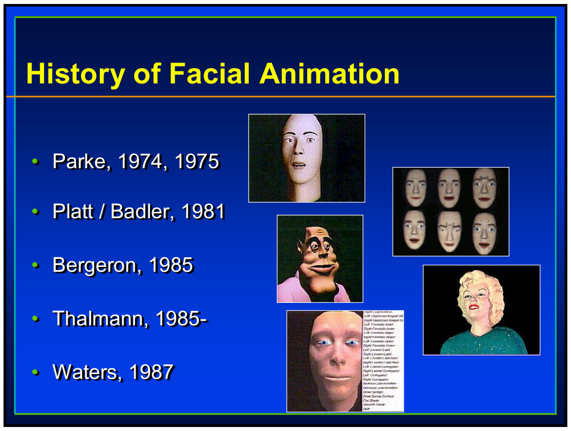
2. Contents

The tutorial notes contain both the slides from the tutorial presentation and some selected publications, which serve as additional background information.

History and Applications of Facial Modeling and Animation

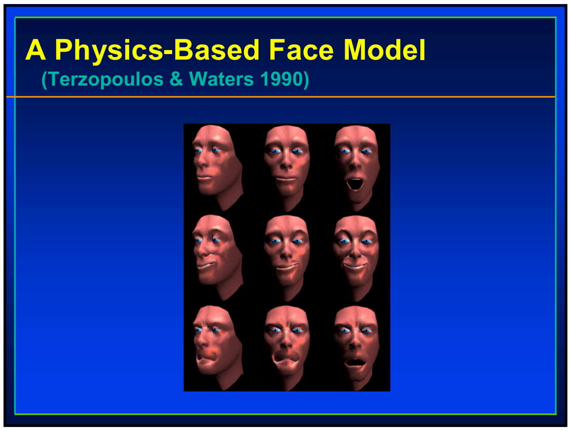
Demetri Terzopoulos

New York University



History of Facial Animation

- Parke, 1974, 1975
- Platt / Badler, 1981
- Bergeron, 1985
- Thalmann, 1985-
- Waters, 1987



A Physics-Based Face Model

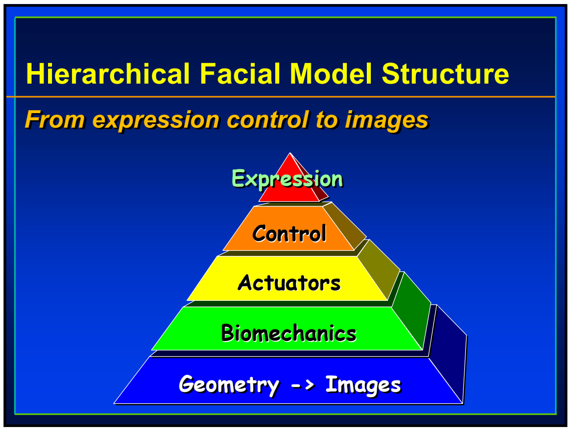
(Terzopoulos & Waters 1990)

A Physics-Based Face Model

(Terzopoulos & Waters 1990)

Hierarchical structure

- Expression: Facial action coding system (FACS)
- Control: Coordinated facial actuator commands
- Muscles: Contractile muscle fibers exert forces
- Physics: Muscle forces deform synthetic tissue
- Geometry: Expressive facial deformations
- Images: Rendering by graphics pipeline



Hierarchical Facial Model Structure

From expression control to images

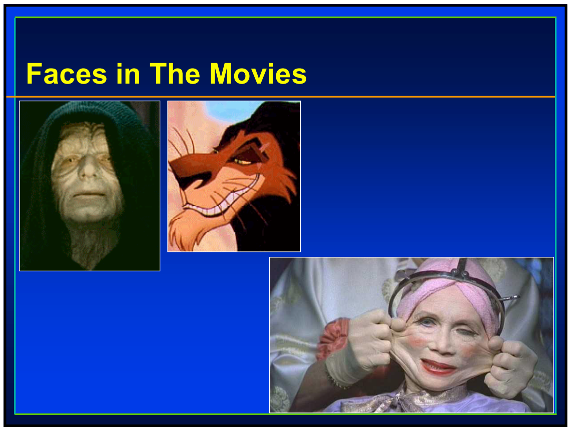
Expression

Control

Actuators

Biomechanics

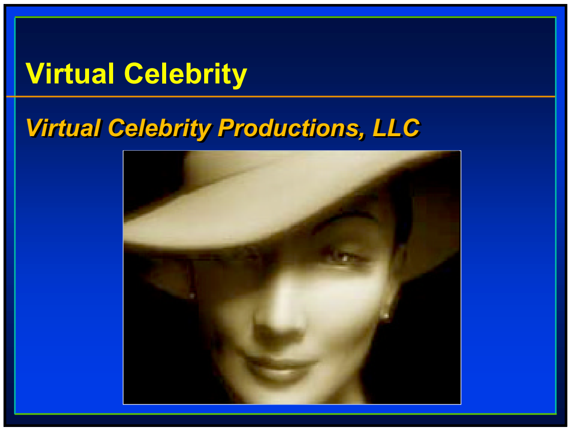
Geometry -> Images



Faces in The Movies

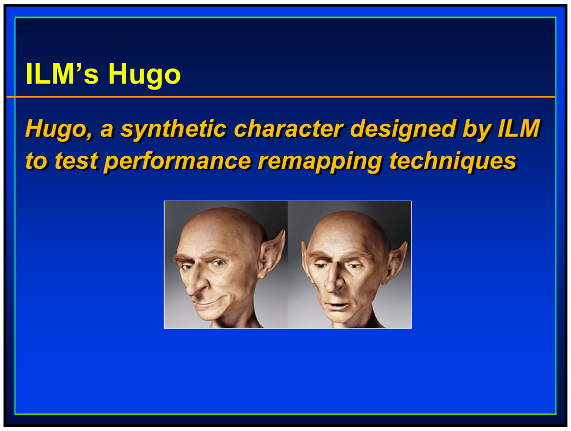
Realistic Facial Modeling

Square USA, Inc.



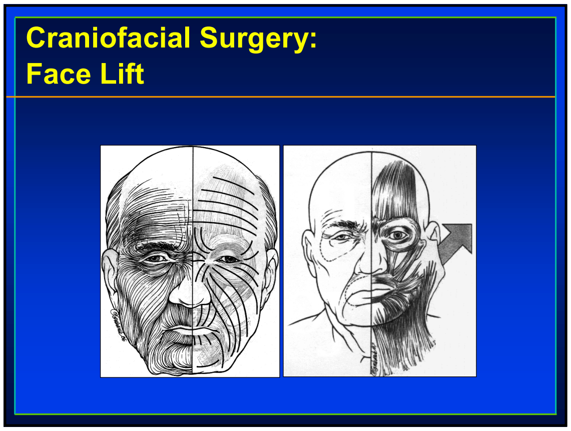
Virtual Celebrity

Virtual Celebrity Productions, LLC

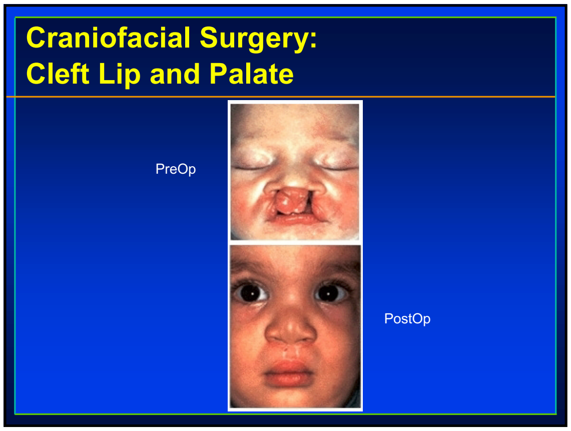


ILM's Hugo

Hugo, a synthetic character designed by ILM to test performance remapping techniques



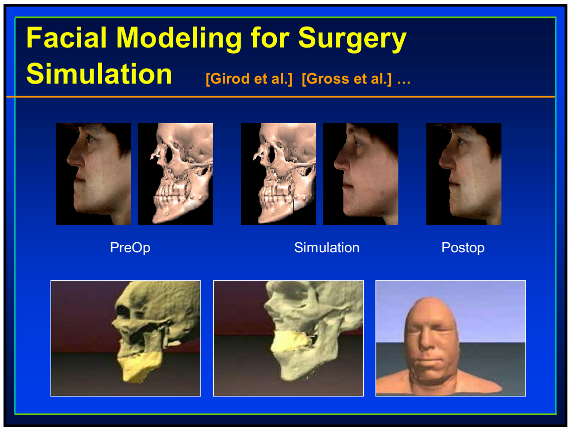
Craniofacial Surgery: Face Lift



Craniofacial Surgery: Cleft Lip and Palate

PreOp

PostOp



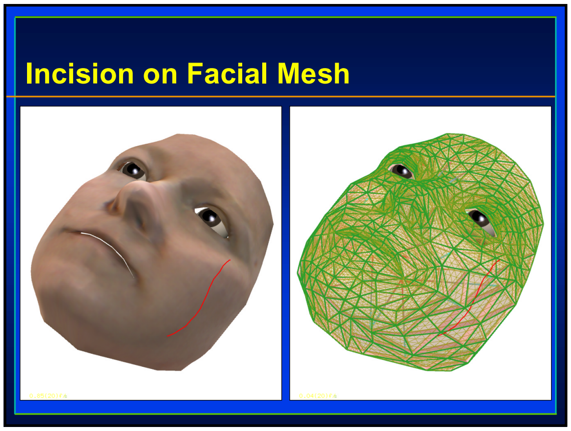
Facial Modeling for Surgery Simulation

[Girod et al.] [Gross et al.] ...

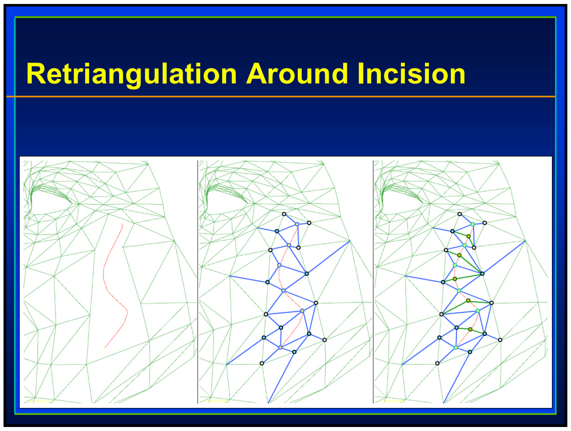
PreOp

Simulation

Postop



Incision on Facial Mesh



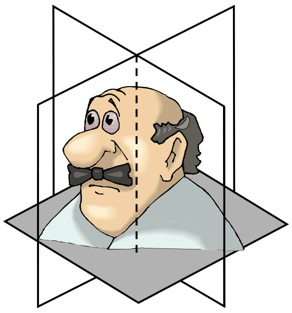
Retriangulation Around Incision

EUROGRAPHICS 2003

Crash course:
Anatomy of the Human Head

Jörg Haber

Terminology



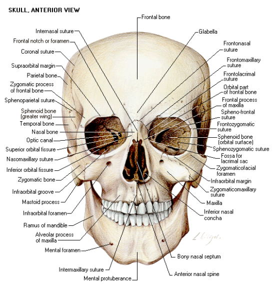
- positions of body parts are described relative to:
  - median (sagittal) plane: vertical plane that divides the body into equal left and right halves; medial / lateral ⇔ closer to / further away from median plane
  - coronal plane: vertical plane that divides the body into front and back halves; (anterior / posterior)
  - transverse (horizontal) plane: any plane perpendicular to both median and coronal planes

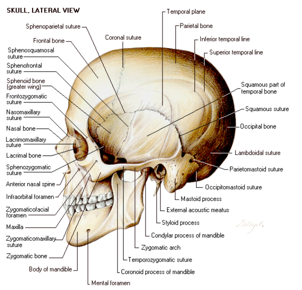
The Human Head

Components of the human head:

- skull (lat. *cranium*)
- facial muscles (lat. *m. faciales et masticatores*)
- skin (lat. *integumentum commune*)
- eyes (lat. *oculi*)
- teeth (lat. *dentes*)
- tongue (lat. *lingua*)

Skull



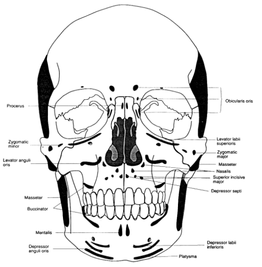


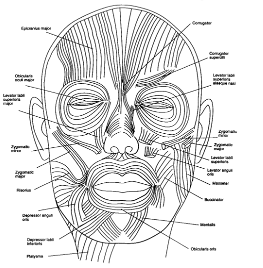
Images: www.humanmuscles.8k.com

Skull

- cranium (lat. *neurocranium*):
  - 7 bones; rigidly connected; lodges and protects brain and eyeballs
  - consists of calvaria and cranial base
- facial skeleton (lat. *viscerocranium*):
  - 15 small bones that surround nasal and oral cavity mosaic-like; only the mandible (lat. *mandibula*) is movable
- bones of the skull are relocatable during birth, ossification completed at the age of 18 ⇒ proportions and shape of the skull change during growth

Facial Muscles (frontal)

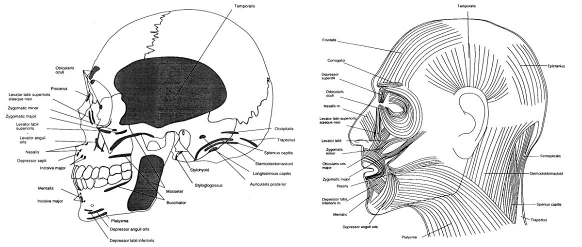




muscles connect a) two bones, b) bone and skin / muscle, or c) two different skin / muscle regions

Images: Parke/Waters: “Computer Facial Animation” (1996)

Facial Muscles (lateral)



Images: Parke/Waters: “Computer Facial Animation” (1996)

Types of Facial Muscles

- sphincters: contract radially towards a center point, e.g. *orbicularis oris*, *orbicularis oculi*
- linear (parallel) muscles: contract longitudinally towards their origin, e.g. *levator labii sup.*, *zygomaticus minor/major*
- sheet muscles: composed of several linear muscles side-by-side, e.g. *frontalis*

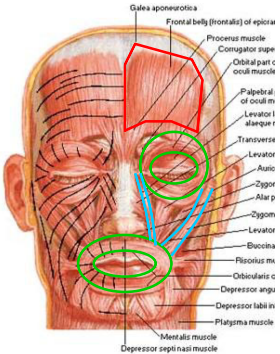
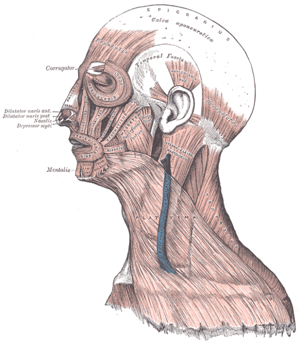


Image: www.humanmuscles.8k.com

Facial Muscles

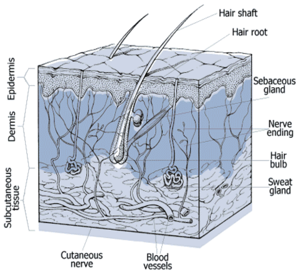


Three groups:

- m. of facial expression: two layers (superficial and deep)
- m. of mastication: movement of the mandible
- epicranius: tension / relaxation of facial skin

Image: Gray: “Anatomy of the Human Body” (1918)

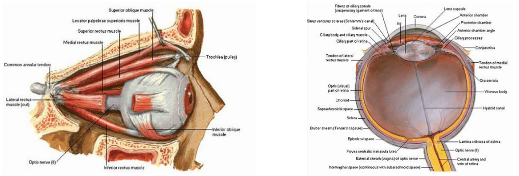
Skin



- epidermis: 0.03-4 mm thick, no vessels, 5 layers of keratin
- dermis: 0.3-2.4 mm thick, 2 layers of soft connective tissue containing elastin fibers, blood and lymphatic vessels, and nerves
- subcutaneous tissue: adipose tissue built from collagen / fat cells, blood vessels, and nerves

Image: www.humanmuscles.8k.com

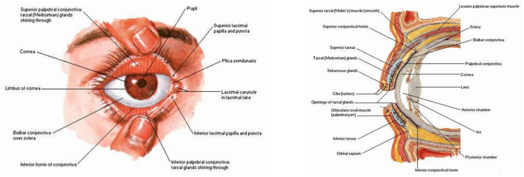
Eyes


Superior oblique muscle
Levator palpebrae superioris muscle
Superior rectus muscle
Medial rectus muscle
Common annular tendon
Trochlea (pulley)
Lateral rectus muscle (cut)
Inferior oblique muscle
Optic nerve (II)
Inferior rectus muscle
Fibers of ciliary zonule (suspensory ligament of lens)
Sinus venosus sclerae (Schlemm's canal)
Scleral spur
Ciliary body and ciliary muscle
Ciliary part of retina
Tendon of lateral rectus muscle
Optic (visual) part of retina
Choroid
Suprachoroidal space
Sclera
Bulbar sheath (Tenon's capsule)
Episcleral space
Fovea centralis in macula lutea
External sheath (vagina) of optic nerve
Intervaginal space (continuous with subarachnoid space)
Lens
Iris
Cornea
Lens capsule
Anterior chamber
Posterior chamber
Anterior chamber angle
Ciliary processes
Conjunctiva
Tendon of medial rectus muscle
Ora serrata
Vitreous body
Hyaloid canal
Lamina cribrosa of sclera
Optic nerve (II)
Central artery and vein of retina

- complex organ consisting of eyeball (lat. *bulbus oculi*) and optic nerve, embedded into the sceletal orbit (lat. *orbita*)
- eyeball composed from lens and viterous body (lat. *corpus vitreum*), enclosed by three concentric layers: sclera / cornea, choroidea / iris, and retina

Images: www.humanmuscles.8k.com

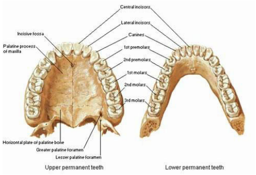
Eyes

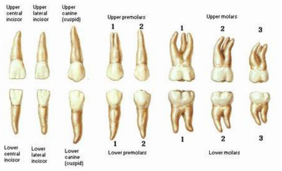

Superior palpebral conjunctiva: tarsal (Meibomian) glands shining through
Pupil
Superior lacrimal papilla and puncta
Cornea
Plica semilunaris
Limbus of cornea
Lacrimal caruncle in lacrimal lake
Bulbar conjunctiva over sclera
Inferior lacrimal papilla and puncta
Inferior fornix of conjunctiva
Inferior palpebral conjunctiva: tarsal glands shining through
Superior tarsal (Muller's) muscle (smooth)
Superior conjunctival fornix
Superior tarsus
Tarsal (Meibomian) glands
Sebaceous glands
Cilia (lashes)
Openings of tarsal glands
Orbicularis oculi muscle (palpebral part)
Inferior tarsus
Orbital septum
Inferior conjunctival fornix
Levator palpebrae superioris muscle
Sclera
Bulbar conjunctiva
Palpebral conjunctiva
Cornea
Lens
Anterior chamber
Iris
Posterior chamber

- eye muscles: alignment of optical axis (external), focussing and adaptation to brightness (internal)
- eyelids, connective tissue: protect from contaminants
- lachrymal: secretion of tears to smooth the cornea, facilitate the motion of the eyeball, and wash away dust particles

Images: www.humanmuscles.8k.com

Teeth

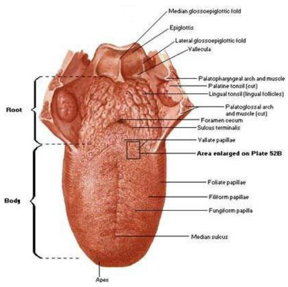




- embedded into upper jaw (lat. *maxilla*) and lower jaw (lat. *mandibula*)
- 20 milk teeth are replaced gradually with 32 permanent teeth starting at the age of about six
- are used to chop up and squelch food, and for articulation

Images: www.humanmuscles.8k.com

Tongue


Median glossoepiglottic fold
Epiglottis
Lateral glossoepiglottic fold
Vallecula
Palatopharyngeal arch and muscle
Palatine tonsil (cut)
Lingual tonsil (lingual follicles)
Root
Palatoglossal arch and muscle (cut)
Foramen cecum
Sulcus terminalis
Vallate papillae
Area enlarged on Plate 52B
Foliate papillae
Body
Filiform papillae
Fungiform papilla
Median sulcus
Apex

- consists of muscle tissue, nerves, blood vessels, and sensory cells (embedded in mucous membrane)
- can alter its shape and position in many ways
- most important sense organ for taste: sweet (tip), salty (front sides), bitter (back)
- support during chewing and swallowing
- use for articulation is learnt

Image: www.humanmuscles.8k.com

All that stuff…

Is it necessary to know all those details?

- it depends on the desired quality / realism of the head model:
  - the more realism you want, the more precisely you have to simulate anatomy
- at least: we need to know about the shape / structure / position of facial components and their interactions
- … so don't be afraid to spend some money on medical textbooks or atlases

Data Acquisition

Volker Blanz

Overview

Motivation

Sources of Data

3D Scanning Techniques

- Criteria for selecting methods
- Main techniques

Processing of 3D data

- Correspondence

Benefits of Real Face Data

- Neutral Head:

  Animate a particular person’s head

- Motion (Expressions or Speech):

  Replicate the identical motion (Motion Capturing)

  Generate new animation using information learned from data (key-frames, modes of variation)

Dimensionality

- 2D Images or Video

  for image-based methods (see afternoon session)

- Surface Data (meshes) embedded in 3D

  standard computer graphics paradigm

- 3D Volumetric

  medical data: CT or MRI

Neutral Head Model

Surface models in 3D from

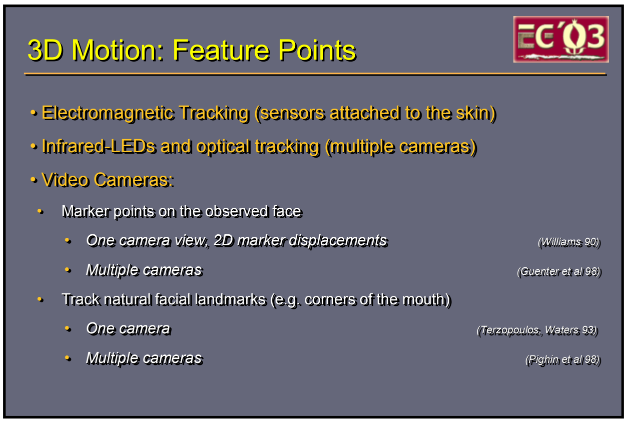
- 3D Scans (Lee, Terzopoulos, Waters 95, Kaehler, Haber, Seidel 01)
- Multiple Camera Views (Lee, Thalmann 98, Pighin et al 98, Fua, Miccio 98)
- Single image (Blanz and Vetter, 99)

3D Motion

- Measure sparse set of features only

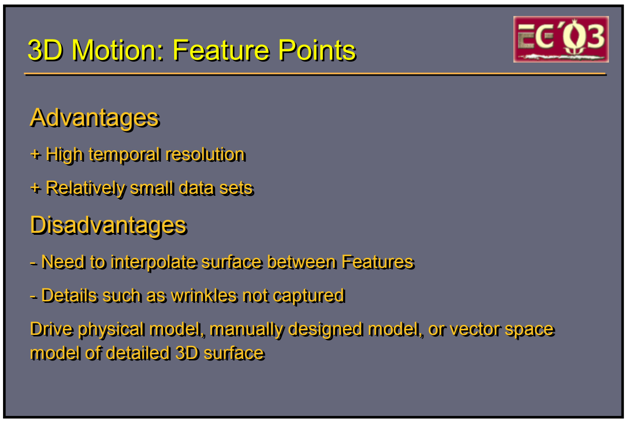
  Interpolate position of mesh vertices in between

- Dense Surface Scans



3D Motion: Feature Points

- Electromagnetic Tracking (sensors attached to the skin)
- Infrared-LEDs and optical tracking (multiple cameras)
- Video Cameras:
  - Marker points on the observed face
    - *One camera view, 2D marker displacements* *(Williams 90)*
    - *Multiple cameras* *(Guenter et al 98)*
  - Track natural facial landmarks (e.g. corners of the mouth)
    - *One camera* *(Terzopoulos, Waters 93)*
    - *Multiple cameras* *(Pighin et al 98)*



3D Motion: Feature Points

Advantages

+ High temporal resolution

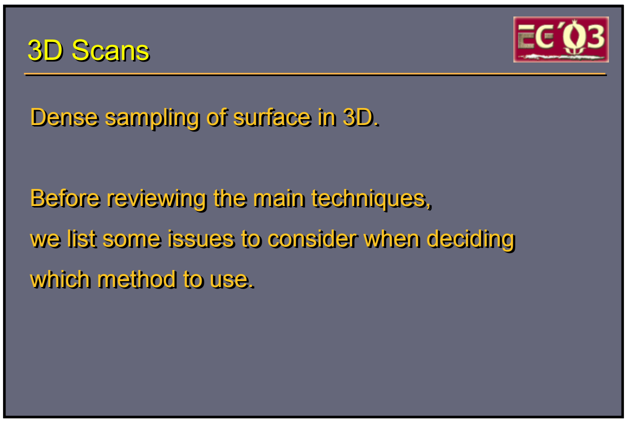
+ Relatively small data sets

Disadvantages

- Need to interpolate surface between Features

- Details such as wrinkles not captured

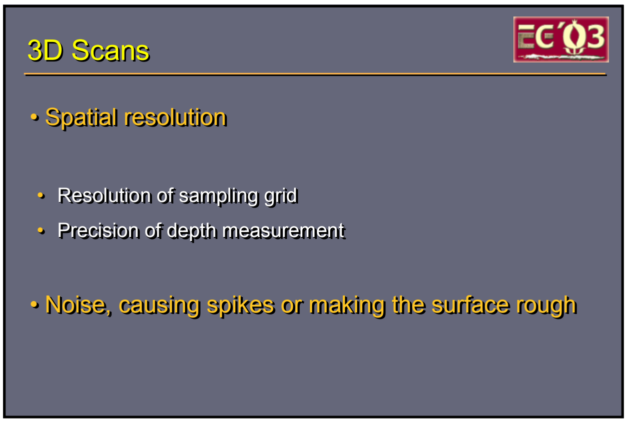
Drive physical model, manually designed model, or vector space model of detailed 3D surface



3D Scans

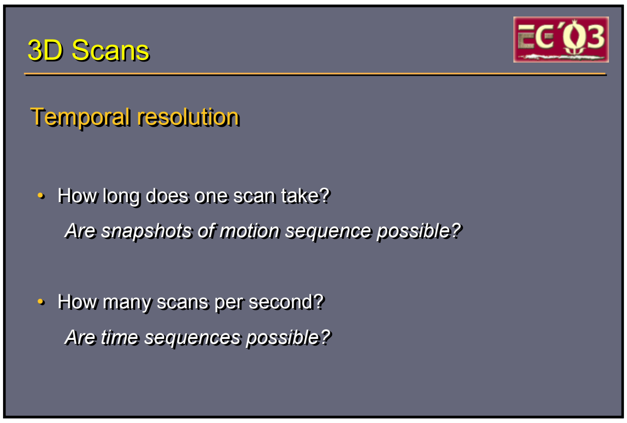
Dense sampling of surface in 3D.

Before reviewing the main techniques,
we list some issues to consider when deciding
which method to use.



3D Scans

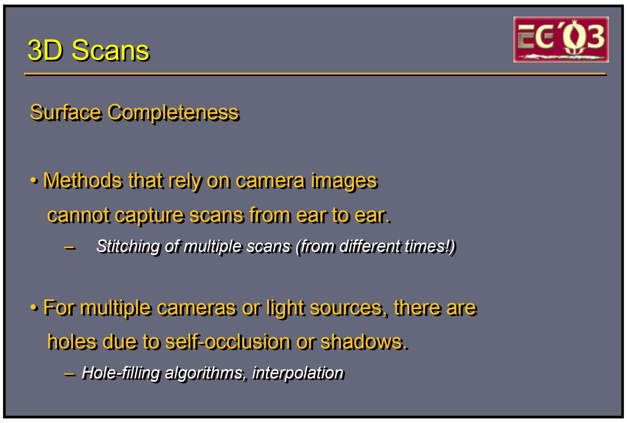
- Spatial resolution
  - Resolution of sampling grid
  - Precision of depth measurement
- Noise, causing spikes or making the surface rough



3D Scans

Temporal resolution

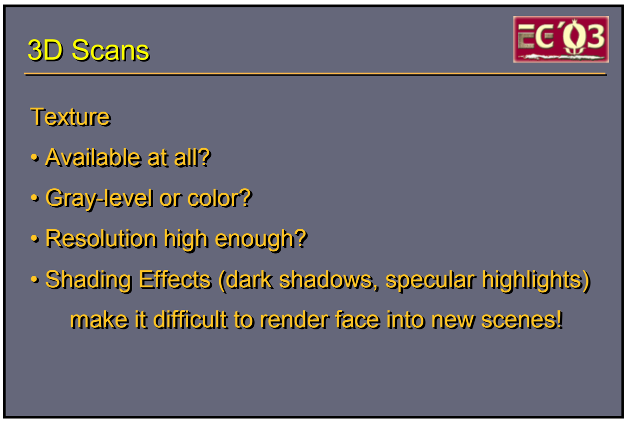
- How long does one scan take?
  Are snapshots of motion sequence possible?
- How many scans per second?
  Are time sequences possible?



3D Scans

Surface Completeness

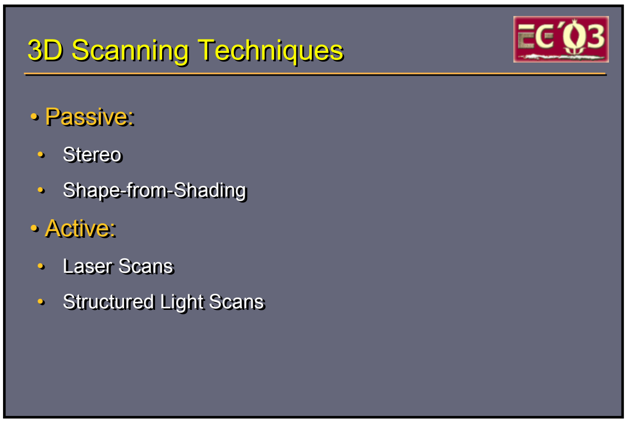
- Methods that rely on camera images
  cannot capture scans from ear to ear.
  – *Stitching of multiple scans (from different times!)*
- For multiple cameras or light sources, there are
  holes due to self-occlusion or shadows.
  – *Hole-filling algorithms, interpolation*



3D Scans

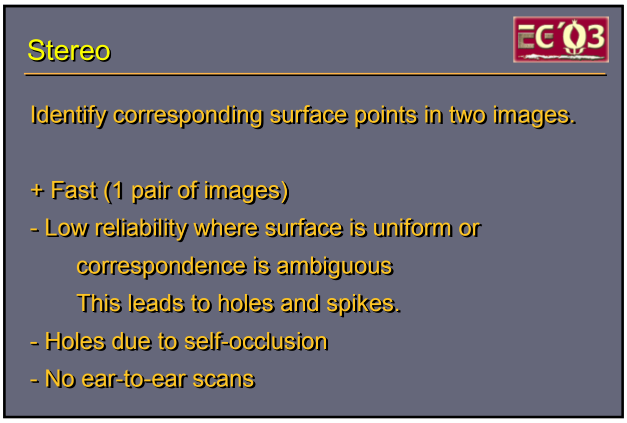
Texture

- Available at all?
- Gray-level or color?
- Resolution high enough?
- Shading Effects (dark shadows, specular highlights)
  make it difficult to render face into new scenes!



3D Scanning Techniques

- Passive:
  - Stereo
  - Shape-from-Shading
- Active:
  - Laser Scans
  - Structured Light Scans



Stereo

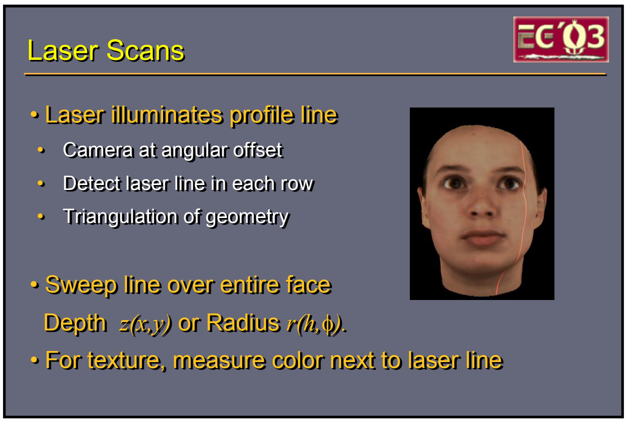
Identify corresponding surface points in two images.

+ Fast (1 pair of images)

- Low reliability where surface is uniform or
correspondence is ambiguous
This leads to holes and spikes.

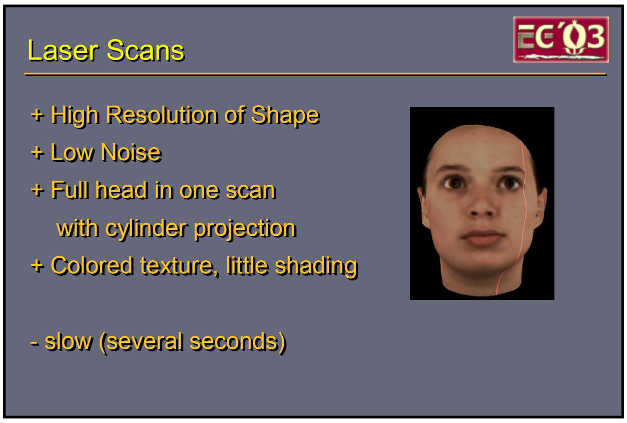
- Holes due to self-occlusion

- No ear-to-ear scans



Laser Scans

- Laser illuminates profile line
  - Camera at angular offset
  - Detect laser line in each row
  - Triangulation of geometry
- Sweep line over entire face
  Depth $z(x,y)$ or Radius $r(h,\phi)$.
- For texture, measure color next to laser line



Laser Scans

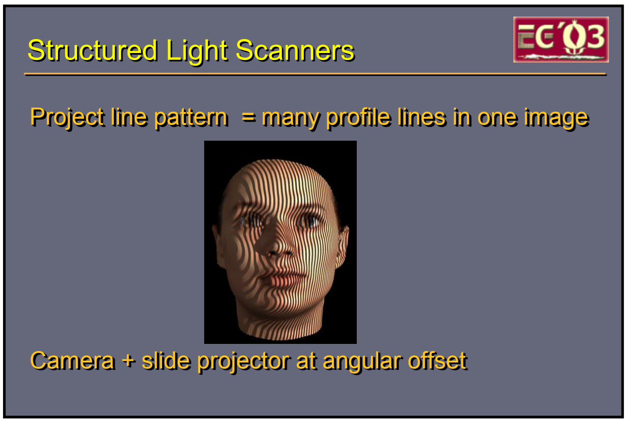
+ High Resolution of Shape

+ Low Noise

+ Full head in one scan
with cylinder projection

+ Colored texture, little shading

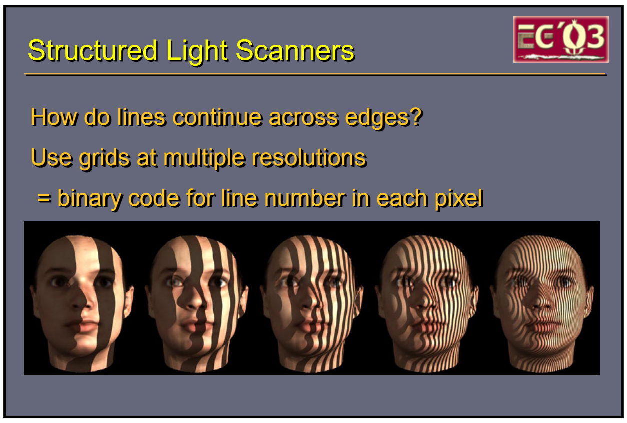
- slow (several seconds)



Structured Light Scanners

Project line pattern = many profile lines in one image

Camera + slide projector at angular offset

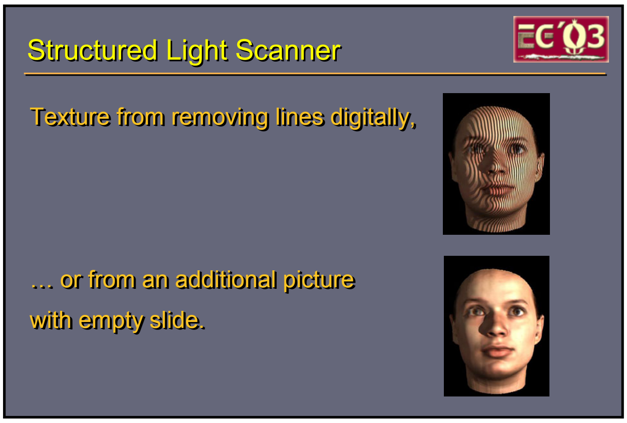


Structured Light Scanners

How do lines continue across edges?

Use grids at multiple resolutions

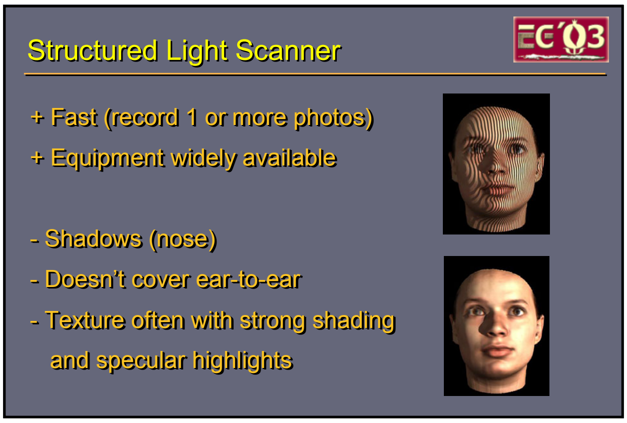
= binary code for line number in each pixel



Structured Light Scanner

Texture from removing lines digitally,

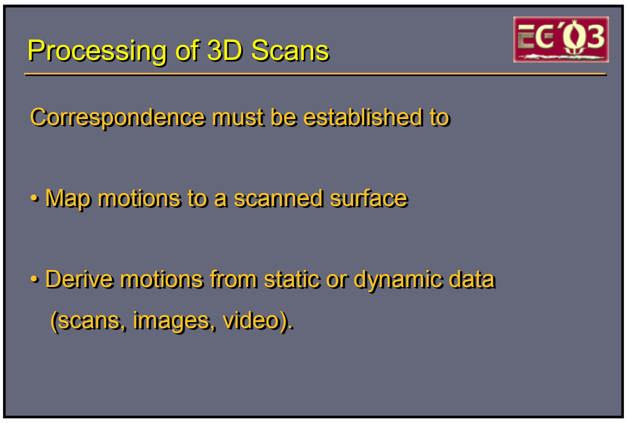
… or from an additional picture with empty slide.



Structured Light Scanner

+ Fast (record 1 or more photos)
+ Equipment widely available

- Shadows (nose)
- Doesn't cover ear-to-ear
- Texture often with strong shading and specular highlights

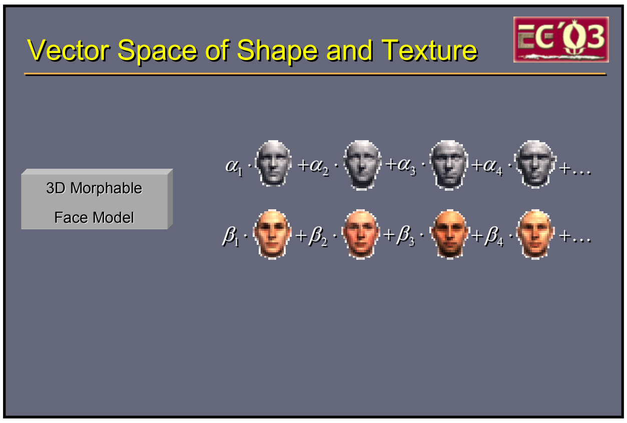


Processing of 3D Scans

Correspondence must be established to

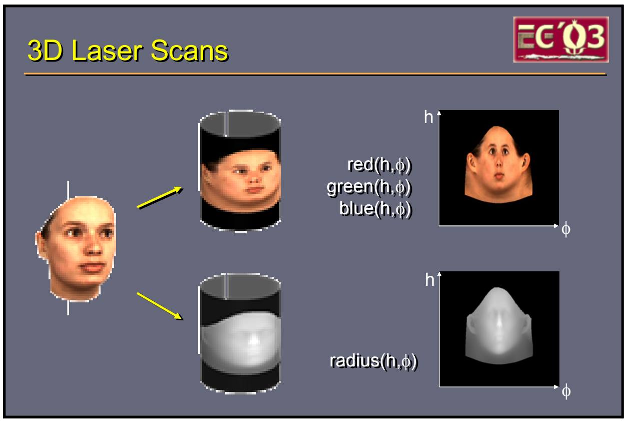
• Map motions to a scanned surface

• Derive motions from static or dynamic data (scans, images, video).



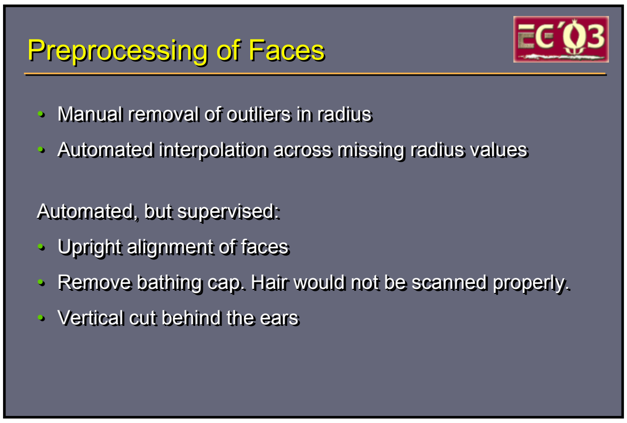
Vector Space of Shape and Texture

3D Morphable Face Model
$\alpha_1 \cdot$ $+\alpha_2 \cdot$ $+\alpha_3 \cdot$ $+\alpha_4 \cdot$ $+\ldots$
$\beta_1 \cdot$ $+\beta_2 \cdot$ $+\beta_3 \cdot$ $+\beta_4 \cdot$ $+\ldots$



3D Laser Scans

h
red(h,ϕ)
green(h,ϕ)
blue(h,ϕ)
ϕ
h
radius(h,ϕ)
ϕ

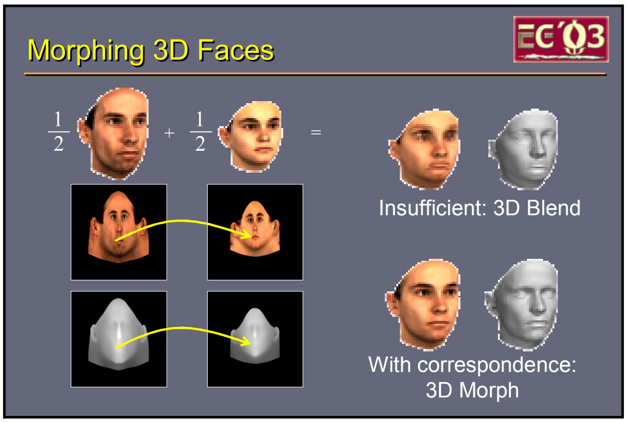


Preprocessing of Faces

- Manual removal of outliers in radius
- Automated interpolation across missing radius values

Automated, but supervised:

- Upright alignment of faces
- Remove bathing cap. Hair would not be scanned properly.
- Vertical cut behind the ears

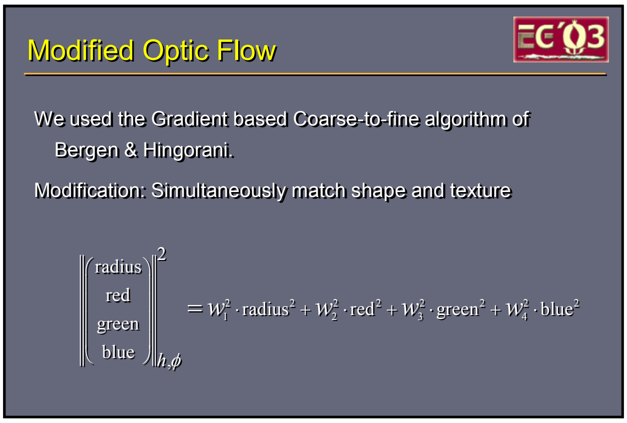


Morphing 3D Faces

$\frac{1}{2}$ + $\frac{1}{2}$ =

Insufficient: 3D Blend

With correspondence:
3D Morph

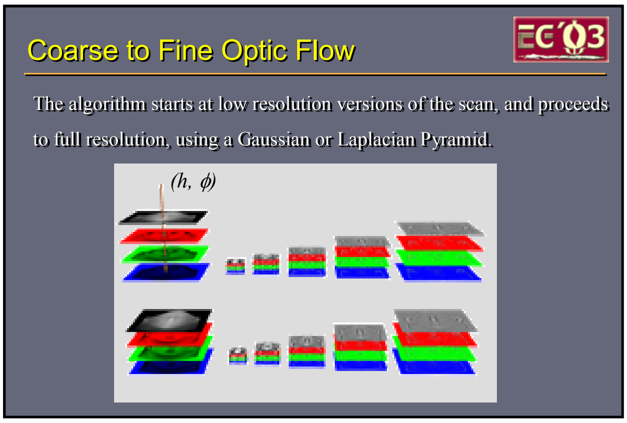


Modified Optic Flow

We used the Gradient based Coarse-to-fine algorithm of Bergen & Hingorani.

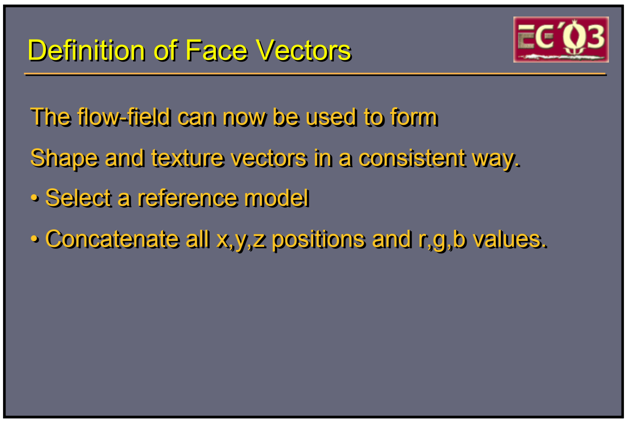
Modification: Simultaneously match shape and texture

$$\left\| \begin{pmatrix} \text{radius} \\ \text{red} \\ \text{green} \\ \text{blue} \end{pmatrix} \right\|_{h,\phi}^{2} = w_1^2 \cdot \text{radius}^2 + w_2^2 \cdot \text{red}^2 + w_3^2 \cdot \text{green}^2 + w_4^2 \cdot \text{blue}^2$$



Coarse to Fine Optic Flow

The algorithm starts at low resolution versions of the scan, and proceeds to full resolution, using a Gaussian or Laplacian Pyramid.

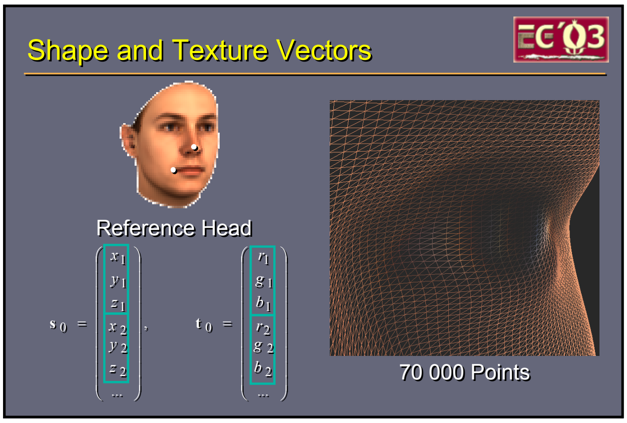


Definition of Face Vectors

The flow-field can now be used to form

Shape and texture vectors in a consistent way.

- Select a reference model
- Concatenate all x,y,z positions and r,g,b values.

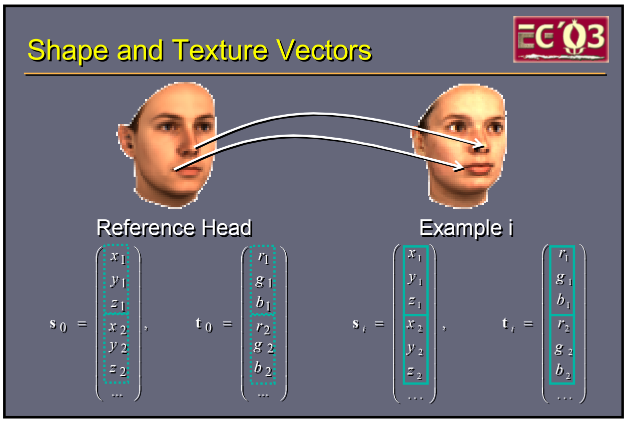


Shape and Texture Vectors

Reference Head

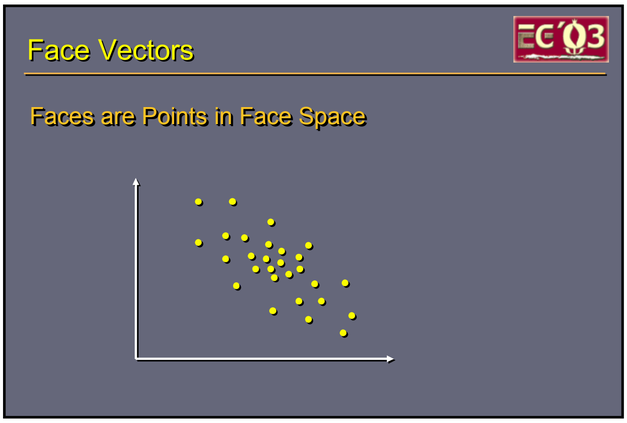
$$\mathbf{s}_0 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \ldots \end{pmatrix}, \quad \mathbf{t}_0 = \begin{pmatrix} r_1 \\ g_1 \\ b_1 \\ r_2 \\ g_2 \\ b_2 \\ \ldots \end{pmatrix}$$

70 000 Points



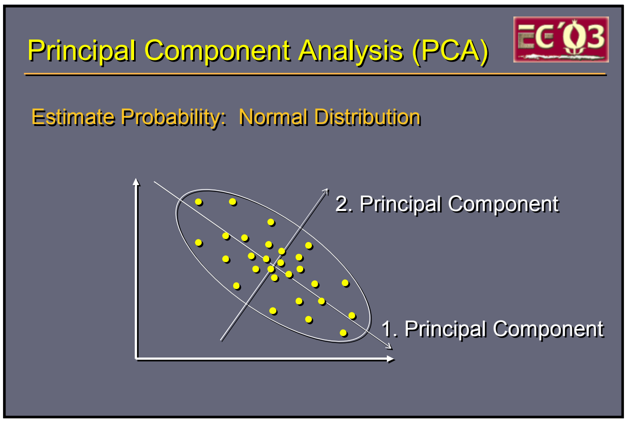
Shape and Texture Vectors

$$\mathbf{s}_0 = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \ldots \end{pmatrix}, \quad \mathbf{t}_0 = \begin{pmatrix} r_1 \\ g_1 \\ b_1 \\ r_2 \\ g_2 \\ b_2 \\ \ldots \end{pmatrix} \qquad \mathbf{s}_i = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \ldots \end{pmatrix}, \quad \mathbf{t}_i = \begin{pmatrix} r_1 \\ g_1 \\ b_1 \\ r_2 \\ g_2 \\ b_2 \\ \ldots \end{pmatrix}$$



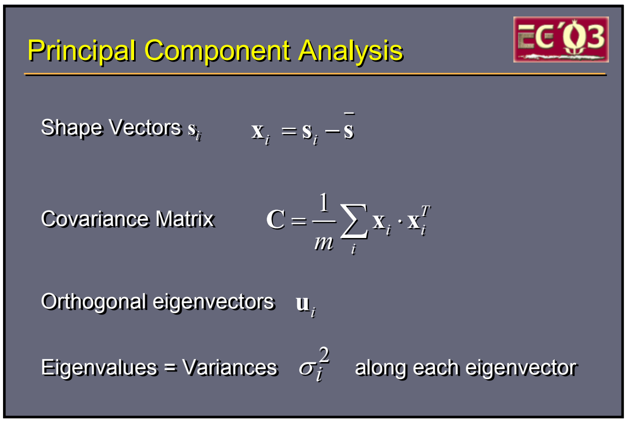
Face Vectors

Faces are Points in Face Space



Principal Component Analysis (PCA)

Estimate Probability: Normal Distribution



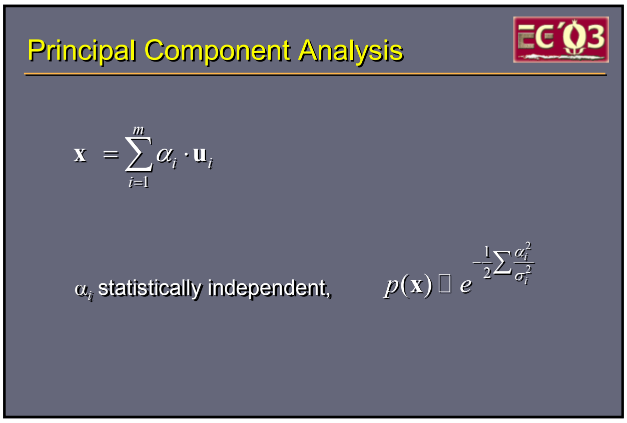
Principal Component Analysis

Shape Vectors $\mathbf{s}_i$ $\quad \mathbf{x}_i = \mathbf{s}_i - \bar{\mathbf{s}}$

Covariance Matrix $\quad \mathbf{C} = \frac{1}{m} \sum_i \mathbf{x}_i \cdot \mathbf{x}_i^T$

Orthogonal eigenvectors $\mathbf{u}_i$

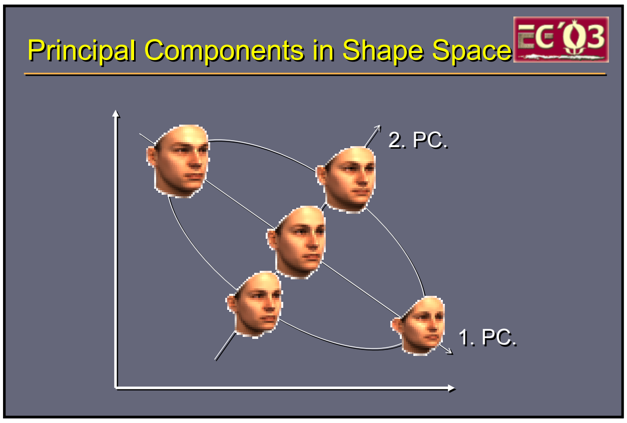
Eigenvalues = Variances σ_i^2 along each eigenvector



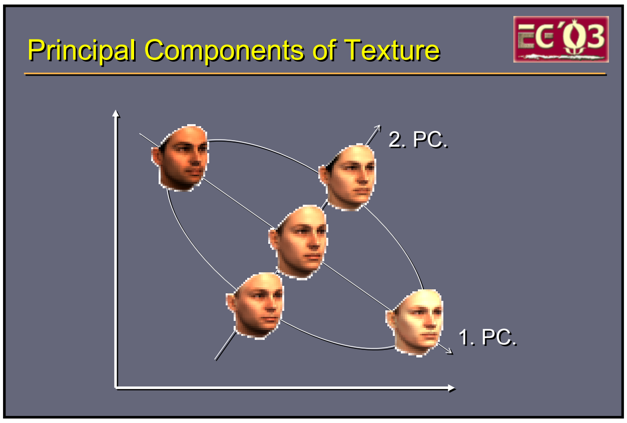
Principal Component Analysis

$$\mathbf{x} = \sum_{i=1}^{m} \alpha_i \cdot \mathbf{u}_i$$

α_i statistically independent, $\quad p(\mathbf{x}) \square e^{-\frac{1}{2}\sum \frac{\alpha_i^2}{\sigma_i^2}}$



Principal Components in Shape Space



Principal Components of Texture

EUROGRAPHICS 2003

Overview:
Facial Animation Techniques

Jörg Haber

Facial Animation

Animation ≠ Animation

- animations with a complete script ("well-known future")
- interactive animations (e.g. computer games)

Different approaches:

- key frame interpolation
- performance-driven animation (→ Demetri)
- direct parameterization (→ Nadia)
- physics-based models (→ Demetri)
- image-based techniques (→ Volker)

Key Frame Interpolation

Completely geometrical approach:

- specify complete face models for given points in time: key frames (key poses, key expressions)
- face models for in-between frames are generated by interpolation

Problematic:

- needs complete face model for each key frame
  ⇒ large amount of data, labor-intensive modeling / acquisition of key frame models
- topology of all key frame models must be identical

Key Frame Interpolation

Types of interpolation:

- convex combination (*linear int.*, *blending*, *morphing*):

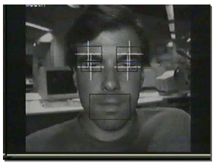
  $$v = \alpha \cdot v_1 + (1 - \alpha) \cdot v_2 \qquad (0 \le \alpha \le 1)$$

  v : scalar or vector (position, color,…)
- non-linear interpolation: e.g. trigonometric functions, splines, …; useful for displaying dynamics (acceleration, slow-down)
- segmental interpolation: different interpolation values / types for independent regions (e.g. eyes, mouth); special treatment for boundaries between regions
  ⇒ decoupling of emotion and speech animation

Performance-driven Animation

Acquisition of animation parameters:

- video camera + software (→ *computer vision*)
- capture head movements, identify eyes and mouth, detect viewing direction and mouth configuration, control synthetic head model with these parameters

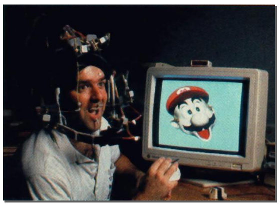


Movies: baback.www.media.mit.edu/~irfan/DFACE.demo/tracking.html

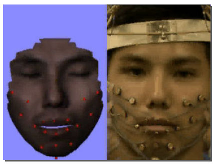
Performance-driven Animation

Acquisition of animation parameters:

- specialized hardware (mechanical / electrical) transfers "deformation" of the human face to a synthetic face model



Virtual Actor system by SimGraphics (1994)



Movie: www.his.atr.co.jp/~kuratate/movie/

Direct Parameterization

Idea:

- perform facial animation using a set of control parameters that manipulate (local) regions / features

What parameterization should be used?

- ideal universal parameterization:
  - small set of intuitive control parameters
  - any possible face with any possible expression can be specified

Direct Parameterization

Image: vismod.www.media.mit.edu/~irfan

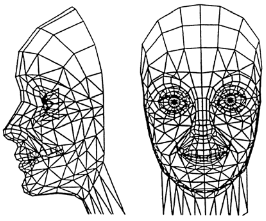
Parametric Models I

- F. I. Parke: "*Parameterized Models for Facial Animation*", IEEE CGA, 2(9):61-68, Nov. 1982
  - 10 control parameters for facial expressions
  - ~20 parameters for definition of facial conformation
- K. Waters: "*A Muscle Model for Animating Three-Dimensional Facial Expression*", SIGGRAPH '87, pp. 17-24, July 1987
  - deforms skin using "muscle vectors"

Parametric Models II

- N. Magnenat-Thalmann et al.: "*Abstract Muscle Action Procedures for Human Face Animation*", The Visual Computer, 3(5):290-297, March 1988
  - pseudo muscles based on empirical models
  - muscle actions are (complex) combinations of FACS action units
- J. E. Chadwick et al.: "*Layered Construction for Deformable Animated Characters*", SIGGRAPH '89, pp. 243-252, July 1989
  - freeform deformations (FFD), pseudo muscles

Parke's Parametric Face Model



- polygonal face mesh (~300 triangles + quads), symmetrical, edges aligned to facial feature lines
- two types of parameters:
  - 10 expression parameters
  - about 20 conformation parameters
- five different ways how parameters modify facial geometry

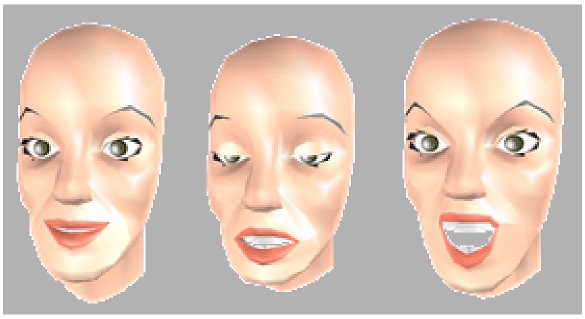
Parke: Expression Parameters

- eyes:
  - dilation of pupils, opening / closing of eyelids, position and shape of eyebrows, viewing direction
- mouth:
  - rotation of mandible, width and shape of the mouth, position of upper lip, position of mouth corners
- additional parameters (suggested):
  - head rotation, size of nostrils

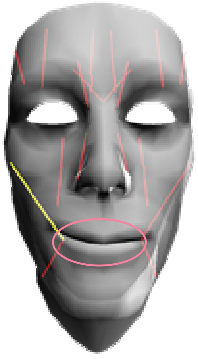
Parke: Conformation Parameters

- aspect ratio of the face
- length and shape of the neck
- shape (= relative position of assigned vertices) of chin, forehead, cheeks, and cheekbones
- size of eyelids, eyeballs, iris; position of the eyes
- jaw width
- length of the nose; width of nose bridge and nostril
- relative size of chin, forehead, and mouth-nose-eyes-part w.r.t. remaining face parts
- color of skin, eyebrows, iris, and lips

Parke: Results



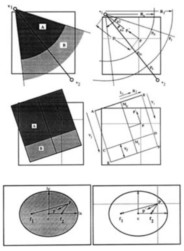
The Face Model by Waters



- polygonal face mesh: 201 quads + 35 triangles
- 10 different muscles:
  - 9 linear muscles (symmetrical left/right)
  - 1 sphincter (*orbicularis oris*)
- additional parameters:
  - jaw rotation
  - viewing direction
  - opening of eyelids

Waters: Muscle Vectors

- muscles are represented by muscle vectors, which describe the effect of muscle contraction on the geometry of the skin surface
- muscle vectors are composed of:
  - a point of attachment and a direction (for linear muscles)
  - a line of attachment and a direction (for sheet muscles)
  - a center point and two semi-axes defining an ellipse (for sphincters)



Images: Waters: “A Muscle Model for Animating Three-Dimensional Facial Expression” (1987)

Physics-based Models

Idea:

- represent and manipulate expressions based on physical characteristics of skin tissue and muscles

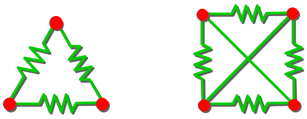
Real anatomy is too complex!

- to date, no facial animation system represents and simulates the complete, detailed anatomy of the human head
- reduce complexity to obtain animatable model
- need to build appropriate models for muscles and skin tissue

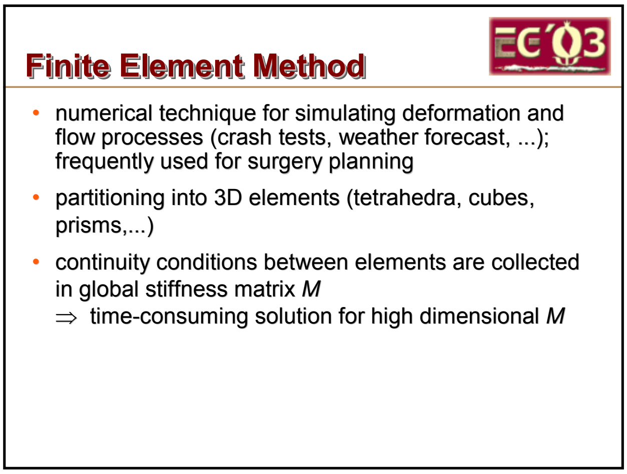
Mass-Spring Networks

- common technique for simulating dynamic behavior of skin tissue
- vertices = mass points, edges = springs
- Lagrangian equations of motion are integrated through time using numerical algorithms
- several variants with multiple layers of mass-spring networks (2D or 3D)

2D:



3D: tetrahedron cube



Finite Element Method

- numerical technique for simulating deformation and flow processes (crash tests, weather forecast, ...); frequently used for surgery planning
- partitioning into 3D elements (tetrahedra, cubes, prisms,...)
- continuity conditions between elements are collected in global stiffness matrix *M*
  ⇒ time-consuming solution for high dimensional *M*

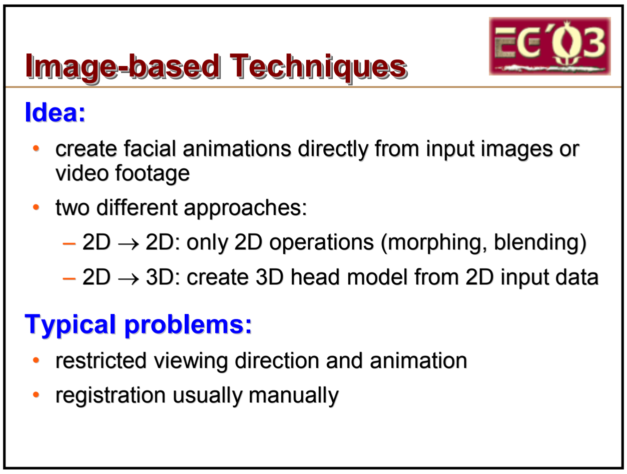


Image-based Techniques

Idea:

- create facial animations directly from input images or video footage
- two different approaches:
  - 2D → 2D: only 2D operations (morphing, blending)
  - 2D → 3D: create 3D head model from 2D input data

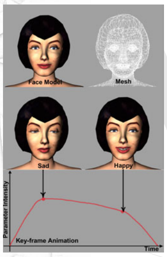
Typical problems:

- restricted viewing direction and animation
- registration usually manually

Parameterized Facial Models

Nadia Magnenat-Thalmann
MIRALab, University of Geneva
thalmann@miralab.unige.ch

Facial Animation : Hierarchy



Step	Technology	Methods
Face Object Definition	Face Modelling, Cloning	Manual, semi-automatic or automatic
Static Expressions Design	Parameterization, Mesh deformation	Manual (GUI), or capture data
Key-frame Animation	Co-articulation for speech, Expression blending	Manual for non-real-time, rule based automatic for real-time

Parameterisation and Deformation

Parameter Set

Deformation

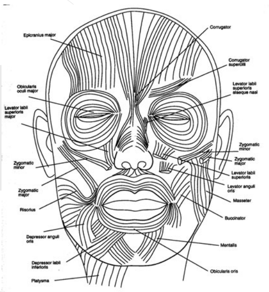
	Parameterisation	Deformation
Definition	Defining an "optimum" set of parameters that can be used to control facial movements	Generating mesh deformation from a given set of values for the parameter set
Importance	For designers/animators	For developers of FA systems
Basic Requirements	Completeness: should incorporate all basic facial movements	Should fully support selected parameter set, resulting into "realistic" facial movements
Additional Requirements	Easy to use, orthogonal (no redundancy)	Speed, ease of implementation

Facial Muscles

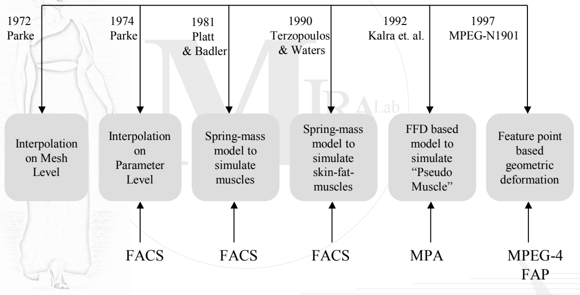
Natural basis to design Parameterisation scheme

Guide to implement deformation methods

Helpful for evaluation of facial animation systems



What is the progress? Parameterisation and Deformation


1972 Parke

1974 Parke

1981 Platt & Badler

1990 Terzopoulos & Waters

1992 Kalra et. al.

1997 MPEG-N1901

Interpolation on Mesh Level

Interpolation on Parameter Level

Spring-mass model to simulate muscles

Spring-mass model to simulate skin-fat-muscles

FFD based model to simulate "Pseudo Muscle"

Feature point based geometric deformation

FACS

FACS

FACS

MPA

MPEG-4 FAP

Parameterisation : Facial Action Coding System

- Initially intended only for facial action description and not for animation
- The system describes the most basic facial muscle actions and their effect on facial expression
- All the muscle actions that can be controlled independently are included
- A set of all possible basic *action units* (AUs) performable by human face and visually distinguishable
- Examples: Inner brow raiser, Lip corner puller, Jaw Drop, Nostril Dilator

P Ekman and W V Frisen, "Facial Action Coding System," Investigator's Guide Part II, Consulting Psychologists Press Inc., 1978.

Parameterisation : Minimal Perceptible Action

- Inspired from FACS
- Supports non-symmetric movements e.g. “lower_left_cornerlip”
- Support for non facial actions e.g. “nod head”, “roll head”
- Support for more detailed actions in mouth region e.g. “pull_midlips”
- Overall, more suitable to develop facial deformation, has possibility to control finer movements than in FACS

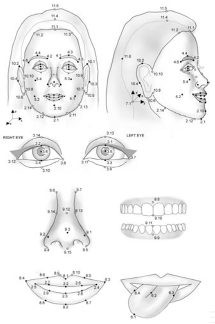
P. Kalra, A. Mangili, N. M. Thalmann, D. Thalmann, Simulation of Facial Muscle Actions Based on Rational Free From Deformations, Eurographics 1992, vol. 11(3), pp. 59-69

Parameterisation : MPEG-4 FAP

Feature Points defined on the Specific locations of the face

Animation defined by the displacements of these Feature Points from neutral position

MPEG-4 does not specify how deformation should be implemented



M. Escher, I. Pandzic, and N. Magnenat-Thalmann. Facial Animation and Deformation for MPEG-4 , *Proc. Computer Animation'98*, 1998

Deformation : Shape Interpolation

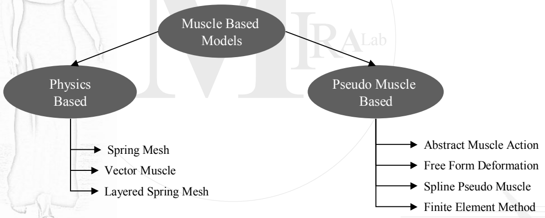
- Earliest approach for facial animation
- Define facial poses by vertex geometry manipulation
- Apply interpolation (linear or non-linear)at vertex level
- Data intensive, model dependant, tedious

F. I. Parke, Keith Waters, Computer Facial Animation, 1996

Deformation : Muscle Based Models

Ability to manipulate facial geometry based on simulating the characteristics of the facial muscles

Mostly using FACS (or variant) as parameters, as FACS are muscle based

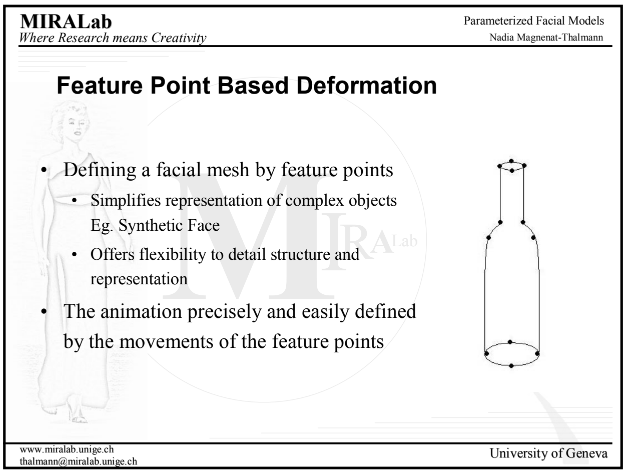

Muscle Based Models
Physics Based
Pseudo Muscle Based
Spring Mesh
Vector Muscle
Layered Spring Mesh
Abstract Muscle Action
Free Form Deformation
Spline Pseudo Muscle
Finite Element Method

A Comparative Look

	Model	Computation	Versatility	Visual Realism	User Control	Parameter Scheme
Physical Based Muscle	Spring	Medium	Low	Low-Med	Medium	FACS
	Vector	Medium	Medium	Med-High	Medium	FASC
	Layered	High	Med-High	Med-High	Hard	FACS
Pseudo Muscle	AMA	Low	Low	Medium	Easy	-
	Spline	Low-Med	Medium	Medium	Easy	FACS
	FFD	Low-Med	Medium	Medium	Easy	MPA/FACS
Other	FEM	High	-	High	Hard	FACS
	Feature Point Based	Low	Medium	Medium	Easy	MPEG-4 FAP

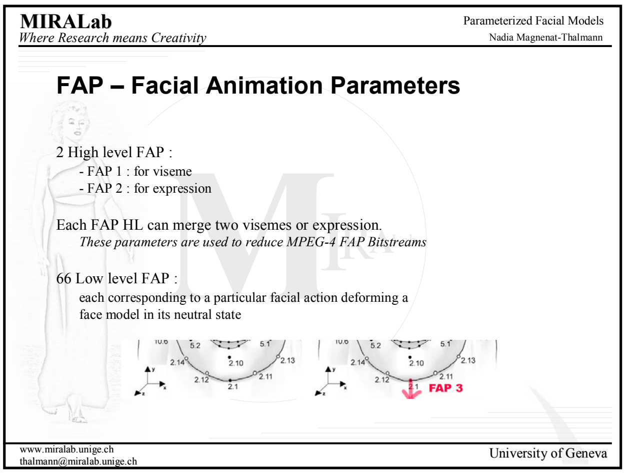
MPEG-4 Facial Animation : A Detailed Look

- Feature point based definition for animatable objects
- MPEG-4 FAPs and Feature Points
- Fast and simple algorithm
- MPEG-4 facial animation for the web



Feature Point Based Deformation

- Defining a facial mesh by feature points
  - Simplifies representation of complex objects Eg. Synthetic Face
  - Offers flexibility to detail structure and representation
- The animation precisely and easily defined by the movements of the feature points

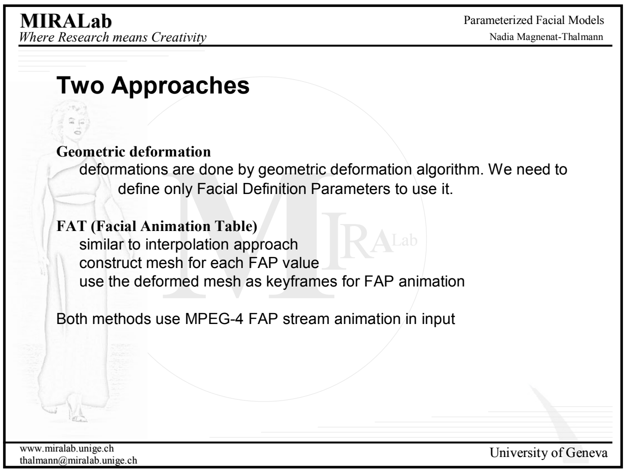


FAP – Facial Animation Parameters

2 High level FAP :
- FAP 1 : for viseme
- FAP 2 : for expression

Each FAP HL can merge two visemes or expression.
These parameters are used to reduce MPEG-4 FAP Bitstreams

66 Low level FAP :
each corresponding to a particular facial action deforming a face model in its neutral state

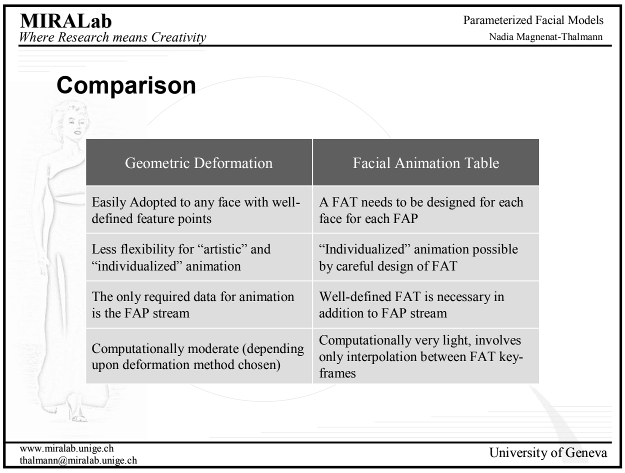


Two Approaches

Geometric deformation
deformations are done by geometric deformation algorithm. We need to define only Facial Definition Parameters to use it.

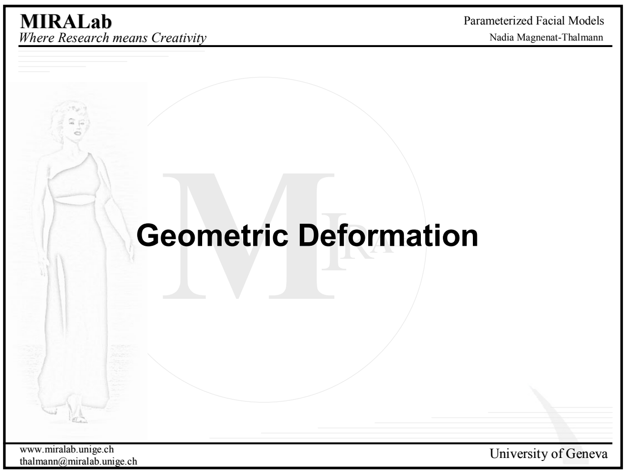
FAT (Facial Animation Table)
similar to interpolation approach
construct mesh for each FAP value
use the deformed mesh as keyframes for FAP animation

Both methods use MPEG-4 FAP stream animation in input

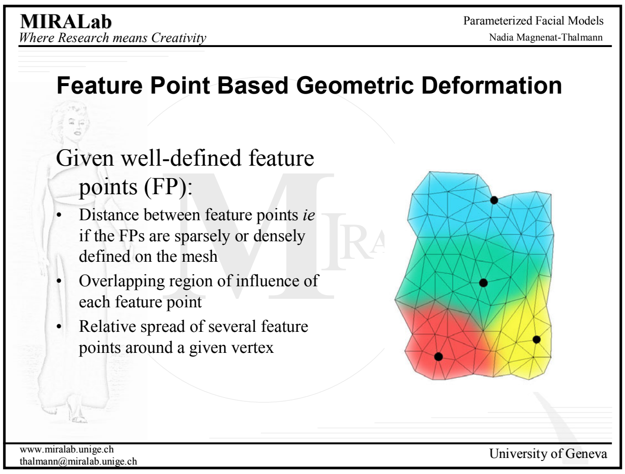


Comparison

Geometric Deformation	Facial Animation Table
Easily Adopted to any face with well-defined feature points	A FAT needs to be designed for each face for each FAP
Less flexibility for "artistic" and "individualized" animation	"Individualized" animation possible by careful design of FAT
The only required data for animation is the FAP stream	Well-defined FAT is necessary in addition to FAP stream
Computationally moderate (depending upon deformation method chosen)	Computationally very light, involves only interpolation between FAT key-frames



Geometric Deformation



Feature Point Based Geometric Deformation

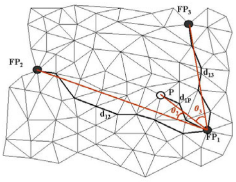
Given well-defined feature points (FP):
- Distance between feature points *ie* if the FPs are sparsely or densely defined on the mesh
- Overlapping region of influence of each feature point
- Relative spread of several feature points around a given vertex

Initialization

- Computing Feature Point Distribution
  Computing d_{12}, d_{13} (*Surface distance*)
- Computing Weights
  For each vertex of the mesh:
  Computing d_{1P}, θ_2, θ_3

$$d = \frac{d_{12}cos(\theta_2) + d_{13}cos(\theta_3)}{cos(\theta_2) + cos(\theta_3)}$$

$$W_{i,P} = sin(\frac{\pi}{2}(1 - \frac{d_{iP}}{d}))$$



Deformation

Displacement of any vertex of the mesh is the weighted sum of all the displacements caused at this vertex by the neighboring feature points

$$D_P = \frac{\sum_{i=0}^{N} \frac{W_{i,P} D_i}{{d_{i,P}}^2}}{\sum_{i=0}^{N} \frac{W_{i,P}}{{d_{i,P}}^2}}$$

N= Number of feature points influencing the vertex P

$W_{i,P}$=Weight of the vertex P for FP_i

D_i=Displacement specified for FP_i

$d_{i,P}$=Distance between vertex P and FP_i

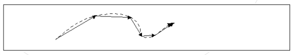
Facial Animation Table Animation

MPEG-4 FAT

Define model spatial deformation as a function of the amplitude of the FAPs

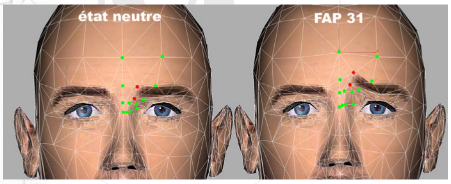
For flexible deformation, piece-wise linear motion trajectories

Vertex n°	**1st interval [I_1, I_2]**	**2nd interval [I_2, I_3]**	**…**
Index 1	displacement D_{11}	displacement D_{12}	…
Index 2	displacement D_{21}	displacement D_{22}	…


An arbitrary motion trajectory is approximated as a piece-wise linear one

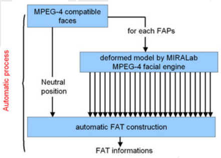
FAT Construction : Artistic Design

An animator designs :
each necessary low level FAPs (with flexible deformation)
high level FAP expressions (joy, sadness, anger, fear, disgust, surprise) and/or visemes

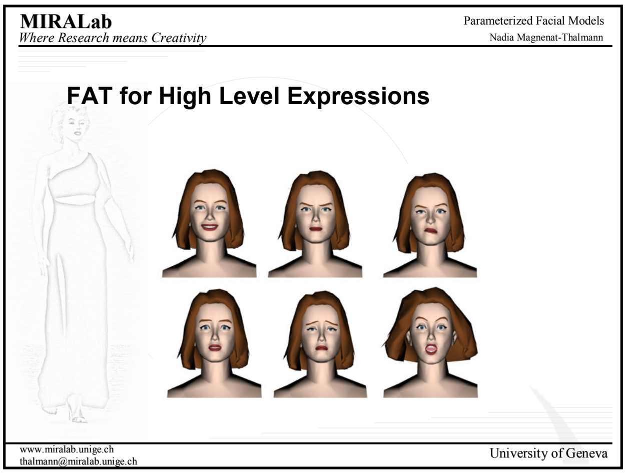


FAT Construction : Automatic Design

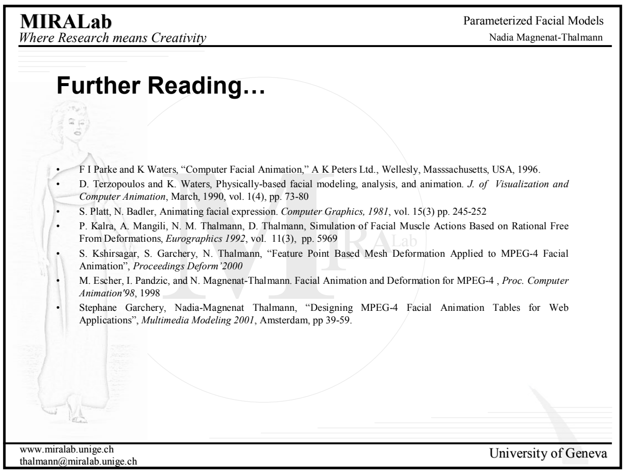
Using Geometric Deformation Technique



Very quick FAT construction (few seconds)
Results into animation very close to MIRALab MPEG-4 facial animation engine



FAT for High Level Expressions



Further Reading…

- F I Parke and K Waters, "Computer Facial Animation," A K Peters Ltd., Wellesly, Masssachusetts, USA, 1996.
- D. Terzopoulos and K. Waters, Physically-based facial modeling, analysis, and animation. *J. of Visualization and Computer Animation*, March, 1990, vol. 1(4), pp. 73-80
- S. Platt, N. Badler, Animating facial expression. *Computer Graphics, 1981*, vol. 15(3) pp. 245-252
- P. Kalra, A. Mangili, N. M. Thalmann, D. Thalmann, Simulation of Facial Muscle Actions Based on Rational Free From Deformations, *Eurographics 1992*, vol. 11(3), pp. 5969
- S. Kshirsagar, S. Garchery, N. Thalmann, "Feature Point Based Mesh Deformation Applied to MPEG-4 Facial Animation", *Proceedings Deform'2000*
- M. Escher, I. Pandzic, and N. Magnenat-Thalmann. Facial Animation and Deformation for MPEG-4 , *Proc. Computer Animation'98*, 1998
- Stephane Garchery, Nadia-Magnenat Thalmann, "Designing MPEG-4 Facial Animation Tables for Web Applications", *Multimedia Modeling 2001*, Amsterdam, pp 39-59.

FEATURE POINT BASED MESH DEFORMATION APPLIED TO MPEG-4 FACIAL ANIMATION

Sumedha Kshirsagar, Stephane Garchery, Nadia Magnenat-Thalmann

MIRALab, CUI
University of Geneva
24 rue du General Dufour
CH-1211 Geneva
Switzerland
{sumedha,garchery,thalmann}@miralab.unige.ch

ABSTRACT

Robustness and speed are primary considerations when developing deformation methodologies for animatable mesh objects. The goal of this paper is to present such a robust and fast geometric mesh deformation algorithm. The algorithm is feature points based *i.e.* it can be applied to enable the animation of various mesh objects defined by the placement of their feature points. As a specific application, we describe the use of the algorithm for MPEG-4 facial mesh deformation and animation. The MPEG-4 face object is characterized by the Face Definition Parameters (FDP), which are defined by the locations of the key feature points on the face. The MPEG-4 compatible facial animation system developed using this algorithm can be effectively used for real time applications. We extract MPEG-4 Facial Animation Parameters (FAP) using an optical tracking system and apply the results to several synthetic facial mesh objects to assess the results of the deformation algorithm.

Keywords: mesh deformation, real-time facial animation, performance driven animation, optical tracking

1 INTRODUCTION

In this paper, we present a robust, fast, and simple geometric mesh deformation algorithm. A geometric mesh can be characterized by the locations of key feature points. Further, the animation of the mesh can be defined by the displacements of these feature points. The algorithm described here can be applied for animation of such meshes. As a specific application, we describe the use of the algorithm for MPEG-4 facial mesh, which is characterized by the Face Definition Parameters (FDP) [13]. We examine the results of the mesh deformation applied to facial animation by using the Facial Animation Parameters (FAP) obtained from an optical tracking system used for facial feature capture. Later part of this section gives a brief background to define our problem. Section 2 details the mesh deformation algorithm that we have developed and implemented. Use of MPEG-4 standard for facial animation and the adaptation of our mesh deformation algorithm to MPEG-4 compliant synthetic face has been discussed in Section 3. In Section 4, we describe the optical tracking method used to extract MPEG-4 Facial Animation Parameters (FAP) and animation using those parameters with a synthetic head model. We give the conclusions and the future work in Section 5.

There are variety of ways possible to represent animatable objects geometrically. The choice depends on the considerations such as precise shape, effective animation and efficient rendering. For the modeling and animation of deformable objects, detailed knowledge about the geometry and the animation structure of the object is necessary. For animatable human like or cartoon characters, modeling of muscles and soft tissue is a complicated task, highly dependant on the specific character under consideration. The underlying models are often simplified with use of various geometric deformation algorithms depending on the applications. Barr introduced geometric modeling deformations using abstract data manipulation operators creating a useful sculpting metaphor [1]. Bearle applied surface patch descriptions to model smooth character form [2]. Free Form Deformation (FFD) and its variants have been used extensively for a variety of modeling and animation applications [4, 10, 3, 14]. They involve the definition and deformation of a lattice of control points. An object embedded within the lattice is then deformed by defining a mapping from the lattice to the object. FFDs allow volume deformation using control points while keeping the surface continuity. They provide the sculptural flexibility of deformations. FFDs have been successfully used for synthetic objects like face [7] and hand deformation [12]. FFDs have some limitations though. The locations of the control points are not very well controllable with respect to the actual mesh object. Also, the discontinuities or holes in the mesh are difficult to handle as a general case. Recently, Singh *et. al.* [15], proposed a new approach of using *wire* curves to define an object and for shaping its deformation. They illustrated the applications of animating figures with flexible articulations, modeling wrinkled surfaces and stitching geometry together.

In order to define shape and animation of a geometric mesh object, we concentrate on the use of feature points. We assume that the shape of the object is defined by the locations of the predefined feature points on the surface of the mesh. Further, the deformation of the mesh can be completely defined by the movements of these feature points (alternatively referred as control points) from their neutral positions either in absolute or in normalized units. This method of definition and animation provides a concise and efficient way of representing an object. Since the control points lie on the geometric surface, their locations are predictable, unlike in FFD. It is difficult to develop and use a physically based deformation technique for a generalized mesh, as the animation of deformable characters requires specific models which are complicated and difficult to generalize. Moreover, this approach offers the flexibility to the implementation of such an object and its animation.

2 GEOEMTRIC MESH DEFINITION AND DEFORMATION

In this section, we describe in detail the feature point based mesh deformation algorithm. The algorithm is usable on any generic surface mesh. To begin with, the feature points or the control points with movement constraint are defined for a given mesh. A constraint in a direction indicates the behavior of the control point in that direction. For example, if a control point is constrained along the x axis, but not along the y and z axes, means that it still acts as an ordinary vertex of the mesh along the y and z axes. Its movement along these axes will be controlled by the other control points in the vicinity.

Any geometric mesh object is a group of vertices with the topological information. Given a geometric mesh with control point locations, we need to compute the regions influenced by each of the control points. In order to get realistic looking deformation and animation, it is necessary that the mesh has a good definition of the feature points. By good definition, we mean that the control point locations should be defined considering the animation properties and real-life topology of the object under consideration. Once the feature points are well defined, a Voronoi surface diagram can be used to divide the mesh into regions, each controlled by a feature point. However, this may result into a patchy and nonrealistic animation. Thus, each vertex of the mesh should be controlled by not only the nearest feature point, but other feature points in the vicinity. The number of feature points influencing a vertex and the factor by which each feature point influences the movement of this vertex is decided by the following:

1. The distances between the feature points *i.e.* if the feature points are spread densely or sparsely on the mesh
2. The distances between the ordinary (non-feature point) vertices of the mesh and the nearest feature point
3. The relative spread of the feature points around a given vertex

The algorithm is divided into two steps. In the *Initialization* step, the above mentioned information is extracted and the coefficients or *weights* for each of the vertices corresponding to the nearest feature points are calculated. The distance between two points is computed as the sum of the edge lengths encountered while traversing from one point to the other. We call this *surface distance.* This *surface distance* measure is useful to handle holes and discontinuities in the mesh. Mouth and eye openings are typical examples of such holes in the facial mesh models. The *Deformation* step actually takes place during the real-time animation for each frame. Actual displacement of all the vertices of the mesh is computed from the displacement of the feature points.

2.1 Initialization

The initialization can further be divided into two substeps.

2.1.1 Computing Feature Point Distribution

In this step, the information about all the neighboring feature points for each of the feature point is extracted. The mesh is traversed starting from each feature point, advancing only one step in all the possible directions at a time, thus growing a mesh region for each feature point, called *feature point region*. Neighboring feature points are those feature points that have a common *feature point region* boundary. As a result, for each feature point defined on the mesh surface, we get a list of the neighboring feature points with *surface distances* between them. This information is further used in the next step which actually calculates the weights associated with each feature point for all the vertices.

2.1.2 Computing Weights

The goal of this step is to extract possible overlapping influence regions for each feature point and to compute the corresponding weight for deformation for all the vertices in this influence region. Consider a general surface mesh as shown in Figure 1. During the process of mesh traversal starting from the feature points, assume that the vertex P is approached from a feature point FP_1. FP_1 is added to the list of the influencing feature points of P. From the information extracted in the previous step of mesh traversal, FP_2, and FP_3 are the neighboring feature points of FP_1. FP_2 and FP_3 are chosen such that the angles θ_2 and θ_3 are the smallest of all the angles θ_i for neighboring feature points FP_i of FP_1. Also,

$$\theta_2 < \frac{\pi}{2}, \theta_3 < \frac{\pi}{2} \tag{1}$$

The surface distances of the vertex from these feature points are respectively d_{1P}, d_{12} and d_{13} as shown in the figure. While computing the weight of FP_1 at P, we consider the effect of the presence of the other neighboring feature points namely FP_2 and FP_3 at P. For this, we compute the following weighted sum d:

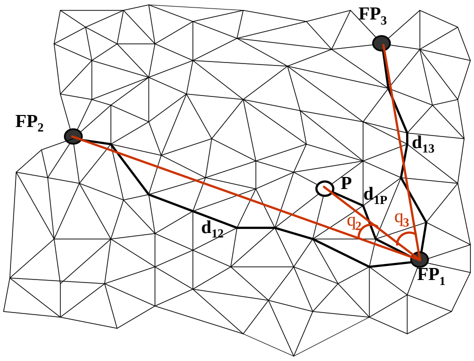


Figure 1: Computing Weights for Animation

$$d = \frac{d_{12}cos(\theta_2) + d_{13}cos(\theta_3)}{cos(\theta_2) + cos(\theta_3)} \tag{2}$$

Thus, d is the weighted sum of the distances d_{12} and d_{13}. The feature point in a smaller angular distance from the FP_1 is assigned a higher value of weight. If there is only one neighboring feature point of FP_1 such that $\theta_2 < \frac{\pi}{2}$, then d is simply computed as $\frac{d_{12}}{cos\theta_2}$.

We compute the weight assigned to the point P for the deformation due to movement of FP_1 as:

$$W_{1,P} = sin(\frac{\pi}{2}(1 - \frac{d_{1P}}{d})) \tag{3}$$

or more generally

$$W_{i,P} = sin(\frac{\pi}{2}(1 - \frac{d_{iP}}{d})) \tag{4}$$

Thus, point P has a weight for displacement that is inversely proportional to its distance from the nearest feature point FP_1. This determines the local influence of the feature point on the vertices of the mesh. At the same time, nearer the other feature points (FP_2 and FP_3 in this case) to FP_1, less is this weight according to the equation 2 and 3. This determines the global influence of a feature point on the surrounding region, in the presence of other feature points in the vicinity.

It is possible that a vertex is approached by more than one feature point, during the process of mesh traversal. We compute the weight for this feature point following the same procedure (considering again the nearest angular neighbors of the feature point), as long as the angular distance criterion (1) is satisfied, and the *surface distance* $d_{iP} < d$,

d as defined in equation 2. This second criterion ensures that the feature points FP_j whose nearest neighbors are nearer to the vertex P than FP_j are not considered while computing the deformation for vertex P. Thus, for the example taken here, weights will be computed for vertex P for the feature points FP_1 as well as FP_2 and FP_3, provided d_{2P} and d_{3P} are less than d. As a result, we have for each vertex of the mesh, a list of control points influencing it and an associated weight.

We tested the algorithm on simple meshes with different values of limits in equation 1, and different weighting functions in equation 2 and 3. The ones giving the most satisfactory results were chosen. In equation 3, instead of *sine* function, it is possible to use any other suitable mathematical operator. We chose *sine* as it is continuous at the minimum and maximum limits. As will be described later, the algorithm applied for MPEG-4 facial mesh produces satisfactory results for deformation and animation.

2.2 Deformation

Once the weights for the vertices have been computed, the mesh is ready for real-time animation. Note that *Initialization* step is computationally intensive, but carried out only once. The weights computed, take into consideration the distance of a vertex from the feature point and relative spread of the feature points around the vertex. Now, from the displacements of the feature points for animation, we calculate the actual displacement of all the vertices of the mesh. Here, we have to consider the effects caused when two or more feature points move at the same time, influencing the same vertex. We calculate the weighted sum of all the displacements caused at the point P due to all the neighboring feature points. Let $FP_i, \quad i = 1, 2, \cdots, N$ be the control points influencing vertex P of the mesh. Then

1. $D_i =$ the displacement specified for the control point FP_i
2. $W_{i,P} =$ the weight as calculated in the *Initialization* for vertex P associated with the control points
3. $d_{i,P} =$ the corresponding distance between P and FP_i.

The following equation gives the resultant displacement D_P caused at the vertex P

$$D_P = \frac{\sum_{i=0}^{N} \frac{W_{i,P} D_i}{{d_{i,P}}^2}}{\sum_{i=0}^{N} \frac{W_{i,P}}{{d_{i,P}}^2}} \qquad (5)$$

This operation is performed for every frame during the computation of the animation of the mesh.

3 ADAPTATION FOR MPEG-4 FACIAL MESH

The problem of facial animation has been approached from various angles. Muscle based models have been effectively developed for facial animation [14, 17, 16]. The Facial Action Coding System [5] defines high level parameters for facial animation, on which several other systems are based. We use MPEG-4 facial animation standard which defines the face object by locations of specific feature points on the facial mesh. The generalized mesh deformation algorithm discussed in the previous section serves well for such an animation framework. Lavagetto *et.al.* have described an MPEG-4 compatible facial animation engine using a similar mesh deformation technique [8]. However, the important difference is that the wireframe semantics have to be specified *a priori* in their method. The wire frame semantics includes specifying the locations of the feature points and the region influenced by each feature point.

3.1 MPEG-4 Facial Animation

The ISO/IEC JTC1/SC29/WG11 (Moving Pictures Expert Group - MPEG) has formulated the new MPEG-4 standard [13]. SNHC (Synthetic Natural Hybrid Coding) is a subgroup of MPEG-4 that has devised an efficient coding method for graphics models and the compressed transmission of their animation parameters specific to the model type. For faces, the Facial Definition Parameter (FDP) set and the Facial Animation Parameter (FAP) set are designed to encode facial shape and texture, as well as animation of faces reproducing expressions, emotions and speech pronunciation.

The FDPs are defined by the locations of the feature points and are used to customize a given face model to a particular face. They contain 3D feature points such as mouth corners and contours, eye corners, eyebrow ends, etc. FAP is based on the study of minimal facial actions and are closely related to muscle actions. They represent

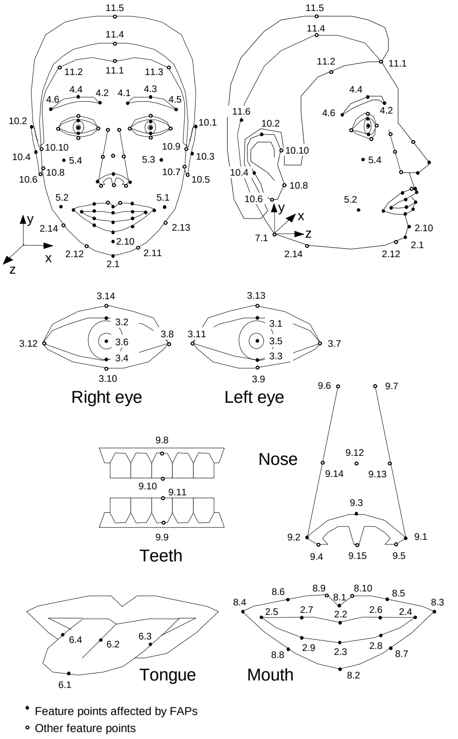


Figure 2: MPEG-4 Facial Feature Points

a complete set of basic facial actions and allow the representation of most natural facial expressions. All parameters involving motion are expressed in terms of the Facial Animation Parameter Units (FAPU). These correspond to fractions of distances between key facial features (e.g. the distance between the eyes). The fractional units are chosen to ensure a sufficient degree of precision. Figure 2 shows the locations of the feature points as defined by the MPEG-4 standard.

3.2 Mesh Deformation using MPEG-4 Feature Points

Given a facial mesh, we can define the locations of the MPEG-4 feature points as per the specification, as shown in Figure 2. Also, for each feature point, we have to define the constraints as defined by the mesh deformation algorithm. For example, the feature point number 2.2 influences the movement of the lips in the y direction. However, in the other two directions, it behaves like an ordinary vertex of the mesh. Thus, its movement is constrained in the y direction by the value of the FAP. Once we define this information, the facial mesh is ready to accept any FAPs and animate the face.

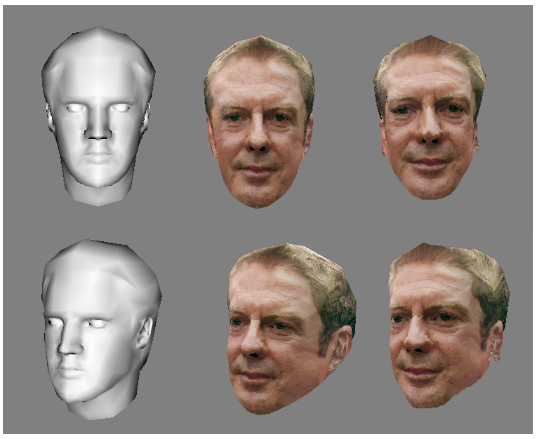


Figure 3: Morphing using the deformation

We also use the same deformation algorithm to deform the facial mesh in order to obtain a new face from a generic mesh. Figure 3 shows the results in two different views. The face on the left side is a generic facial mesh. The face in the middle is acquired using two orthogonal photographs of a person using the technique described in [9]. In this method, the locations of the feature points are extracted from the images and Rational Free Form Deformation (RFFD) is used to deform the the generic face. Appropriate texture mapping is done to add realism. Note that the feature points used for this purpose are as high as 160 in number. We use a subset of these feature points containing 84 points specified by the MPEG-4 standard. We apply the deformation algorithm explained in the previous section to the same generic face using these MPEG-4 feature points to obtain the face on the right. Thus the deformation algorithm applied for 3D morphing of generic head using MPEG-4 feature points generates satisfactory result.

4 OPTICAL TRACKING FOR ANIMATION

Facial feature tracking efforts have ranged from an ordinary video camera with coloured markers to retro-reflective markers and infrared camera to extract directly the 3D position of the markers. We use one such commercially available system (VICON 8) to track the facial expressions and retarget the tracked features to our facial animation engine to examine the results of the deformation algorithm. We use the MPEG-4 feature points corresponding to the FAP values to track the face and extract the FAPs frame by the frame. The next subsection in brief explains the algorithm for extracting the global head rotation and the calculation of the FAP values with the under-

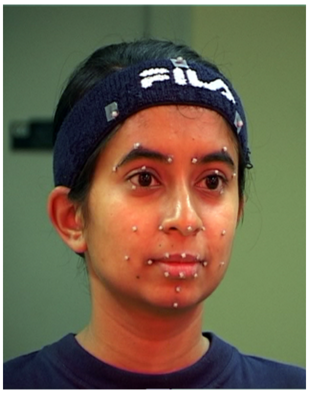


Figure 4: Placement of Markers for Selected MPEG-4 Feature Points

lying assumptions. Subsequently we explain the results obtained with the animation followed by the limitations and scope for improvement. For the capture, we used 6 cameras and 27 markers corresponding to the MPEG-4 feature point locations. 3 additional markers are used for tracking the global orientation of the head. Figure 4 shows the placement of the feature points on the actor's face. It is difficult to obtain the tracking when the markers get too near to each other during facial movements. Hence, the markers corresponding to the feature points along the inner contour of the lips are not used. Instead, the marker positions from the outer lip contour are used as an approximation. Also, the markers on the eyelids are offset so that they do not touch each other during blinks and eye closure. As the FAPs corresponding to these markers specify movement only in vertical direction, this adjustment in the position is justified and does not give rise to error. We get the 3D trajectories for each of the marker points as the output of the tracking system.

4.1 Extracting Global Head Movements

We use 3 markers attached to the head to capture the rigid head movements (the global rotation and translation of the head). We use the improved translation invariant method [11]. Let (p_i, p'_i) be the positions of the points on the surface of the rigid body, observed at two different time instants. For a rigid body motion, the pair of points (p_i, p'_i) obey the following general displacement relationship:

$$p'_i = Rp_i + t \qquad i = 1, 2, \cdots N \tag{6}$$

R is a 3×3 matrix specifying the rotation angle of the rigid body about an axis arbitrarily oriented in the three dimensional space, whereas t represents a translation vector specifying arbitrary shift after rotation. Three non-collinear point correspondences are necessary and sufficient to determine R and t uniquely. With three point correspondences, we get nine non-linear equations while there are six unknown motion parameters. Because the 3D points obtained from the motion capture system are accurate, linear algorithm is sufficient for this application, instead of iterative algorithms based on least square procedure. If two points on the rigid body, p_i and p_{i+1}, which undergo the same transformation, move to p'_i and p'_{i+1} respectively, then

$$p'_i = Rp_i + t \tag{7}$$

and

$$p'_{i+1} = Rp_{i+1} + t \tag{8}$$

Subtraction eliminates the translation t, and using the rigidity constraints yields:

$$\frac{p'_{i+1} - p'_i}{\mid p'_{i+1} - p'_i \mid} = R\frac{p_{i+1} - p_i}{\mid p_{i+1} - p_i \mid} \tag{9}$$

The above equation is defined as:

$$\hat{m}'_i = R\hat{m}_i \tag{10}$$

If the rigid body undergoes a pure translation, these parameters do not change, which means the translation is invariant. After rearranging these three equations, we can solve a 3×3 linear system to get R and afterwards obtain t by substitution in equation 6. In order to find a unique solution, the 3×3 matrix of unit $\hat{m}$ vectors must be of full rank, meaning that the three $\hat{m}$ vectors must be non-coplanar. As a result, four point correspondences are needed. To overcome this problem of supplying the linear method with an extra point correspondence, a "pseudo-correspondence" can be constructed due to the property of rigidity. We find a third $\hat{m}$ vector orthogonal to the two obtained from three points attached to the head. Thus, the system has lower dimension, requiring only three non-collinear rigid points.

Once we extract the global head movements, the motion trajectories of all the feature point markers are compensated for the global movements, and the absolute local displacements for each are calculated. To calculate the MPEG-4 FAP, we also need the FAPU (Facial Animation Parameter Units), which are the distances between the key locations like distance between the lip corners, height of nose. These can be easily calculated from the still frame during initialization.

5 CONCLUSION AND FUTURE WORK

Figure 5 shows the frames of animation depicting different facial expressions on the real face and three different synthetic faces. With the mesh deformation algorithm described here, the computation can be done at as high a frame rate as 70 frames per second for an MPEG-4 compatible facial mesh with 1257 vertices on a 600 MHz Pentium III PC. Depending upon the graphics capabilities, the actual real-time animation performance varies. We can obtain 29 frames per second on the same model and the same PC with Matrix G400 graphics card using Open GL Optimizer for rendering. Thus, the algorithm is well suited for real time MPEG-4 compatible facial animation. We have further used optical tracking to extract the facial features in 3D and obtain the synthetic facial animation to examine the result of the deformation algorithm described here. In order to assess the generality of the mesh deformation algorithm, it needs to be applied to a variety of other synthetic mesh objects.

6 ACKNOWLEDGEMENTS

This work is supported by the EU ACTS SONG project. We are thankful to the MIRALab staff for their valuable support and help. We would like to give special thanks to Dr. Tom Molet for his help with optical tracking system and to Chris Joslin for proof reading this paper.

REFERENCES

[1] A. Barr, "Global and Local Deformations of Solid Primitives", Computer Graphics, Vol. 18, No. 3, July 1984.

[2] V. Bearle, "A Case Study of Flexible Figure Animation", 3-D Character Animation by Computer Course Notes, Siggraph'87.

[3] Y-K. Chang and A. Rockwood. "A generalizad de casteljau approach to 3d free-form deformation" Computer Graphics Proceedings of SIGGRAPH'94, pages 257–260

[4] C. Sabine. "Extended Free-Form Deformation: A Sculpting Tool for 3D Geometric Modeling" Proceedings of SIGGRAPH '90, In Computer Graphics, 24, 4, pages 187–196, August 1990.

[5] E. Friesen WV (1978), Facial Action Coding System: A Technique for the Measurement of Facial Movement. Palo Alto, California: Consulting Psychologists Press.

[6] D. Forsey and R. Bartels, "Hierarchical B-Spline Refinement", Computer Graphics, Vol. 22, No. 4, 1988.

[7] P. Kalra, A. Mangili, N. Magnenat-Thalmann, D. Thalmann, "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", Proc. Eurographics '92, Cambridge, pp. 59-69.

[8] F. Lavagetto, R. Pockaj, "The Facial Animation Engine: towards a high-level interface for the design of MPEG-4 compliant animated faces" IEEE Trans. on Circuits and Systems for Video Technology, Vol. 9, no.2, March 1999.

[9] W. Lee, N. Magnenat-Thalmann, "Fast Head Modeling for Animation", Journal of Image and Vision Computing, Volume 18, Number 4, pp.355-364, Elsevier, 1 March, 2000.

[10] R. MacCracken and K. Joy, Free-form deformation with Lattices of arbitrary topology. Computer Graphics (Proc. of SIGGRAPH'96), pp. 181188, 1996.

[11] W. Martin and J. Aggarwal, "Motion Understanding Robot and Human Vision", Kluwer Academic Publishers, 1988.

[12] L. Moccozet, N. Magnenat-Thalmann, "Dirichlet Free-Form Deformations and their Application to Hand Simulation", Proc. Computer Animation '97, IEEE Computer Society, 1997, pp. 93-102.

[13] Specification of MPEG-4 standard, Moving Picture Experts Group, http://www.cselt.it/mpeg/

[14] F. Parke, "Parameterized Models for Facial Animation", IEEE Computer Graphics and Applications, Vol.2, Nuo. 9, pp 61-68, November 1982

[15] T. Sederberg and S. Parry, "Free Form Deformations of Solid Geometric Models", Computer Graphics, Vol. 20, No. 4, 1986.

[16] K. Singh, E. Fiume, "Wires: A Geometric Deformation Technique", Proc. SIGGRAPH'98.

[17] D. Terzopoulos, K. Waters, "Physically Based Facial Modelling, Analysis and Animation", Journal of visualization and Computer Animation, Vo. 1, No. 2, pp 73-90, 1990.

[18] K. Waters, "A Muscle Model for Animation Three Dimensional Facial Expression", Computer Graphics, Vol. 21, No. 4, pp 17-24, July 1987.

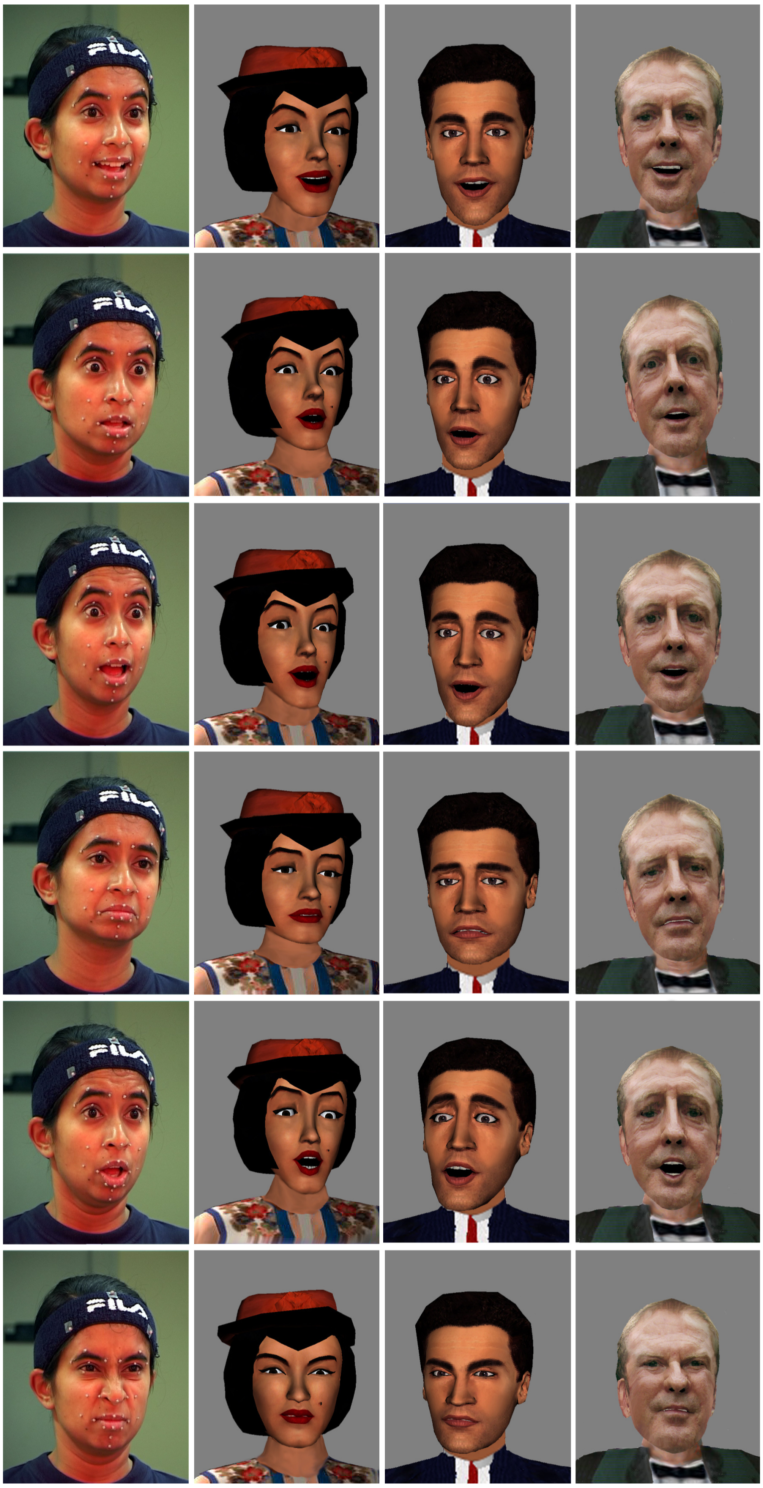
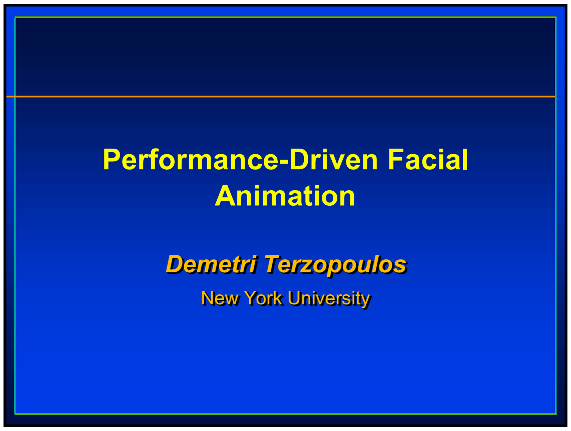


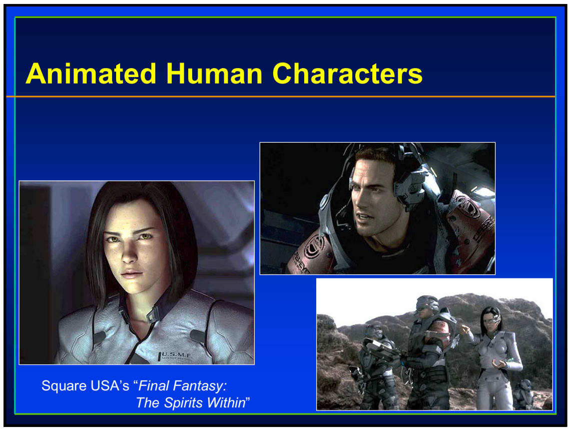
Figure 5: Facial Expressions Extracted by Optical Tracking Reapplied to Different MPEG-4 Faces



Performance-Driven Facial Animation

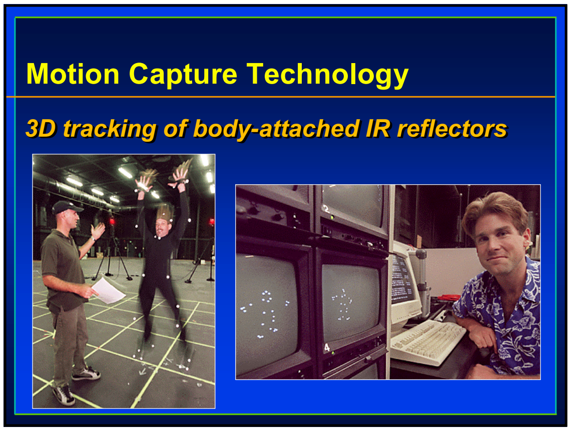
Demetri Terzopoulos

New York University



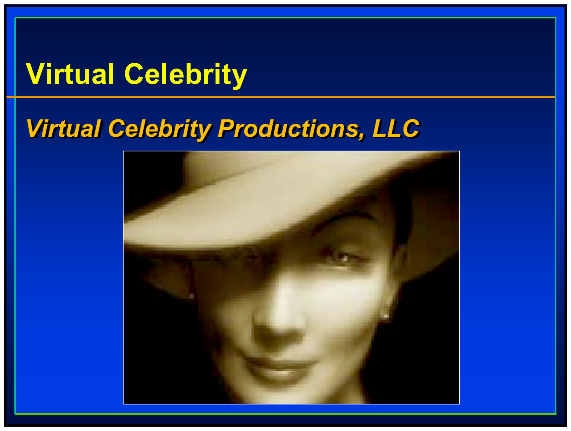
Animated Human Characters

Square USA's "*Final Fantasy: The Spirits Within*"



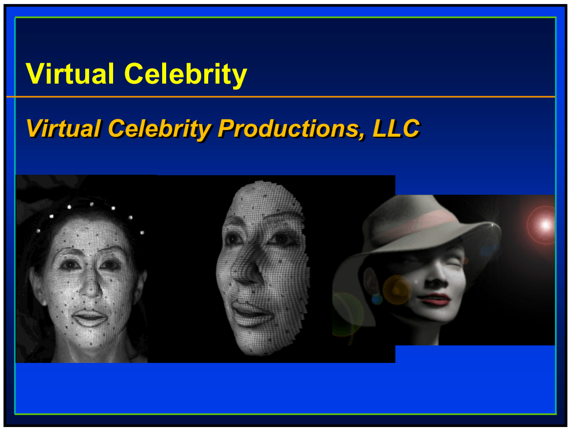
Motion Capture Technology

3D tracking of body-attached IR reflectors



Virtual Celebrity

Virtual Celebrity Productions, LLC



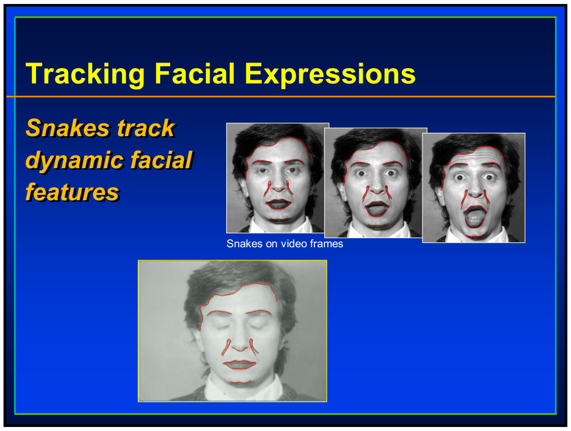
Virtual Celebrity

Virtual Celebrity Productions, LLC

"Eve Solal"

Motion Capture

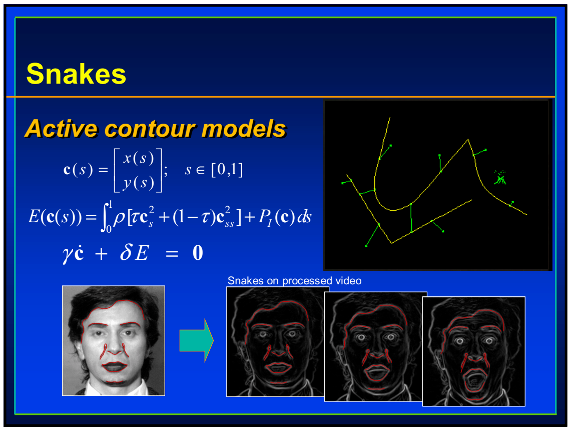
Attitude Studio



Tracking Facial Expressions

Snakes track dynamic facial features

Snakes on video frames



Snakes

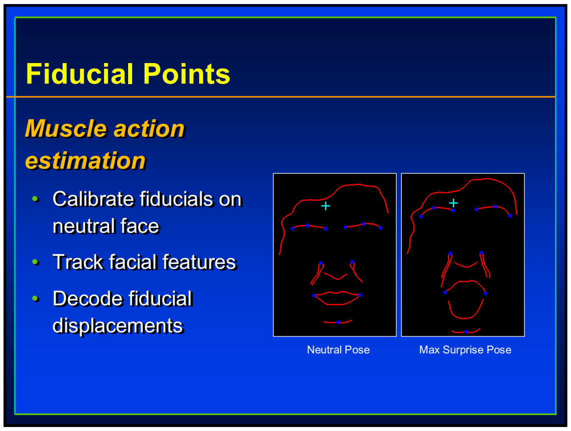
Active contour models

$$\mathbf{c}(s)=\begin{bmatrix}x(s)\\ y(s)\end{bmatrix};\quad s\in[0,1]$$

$$E(\mathbf{c}(s))=\int_0^1\rho[\tau\mathbf{c}_s^2+(1-\tau)\mathbf{c}_{ss}^2]+P_I(\mathbf{c})\,ds$$

$$\gamma\dot{\mathbf{c}}\;+\;\delta E\;=\;\mathbf{0}$$

Snakes on processed video



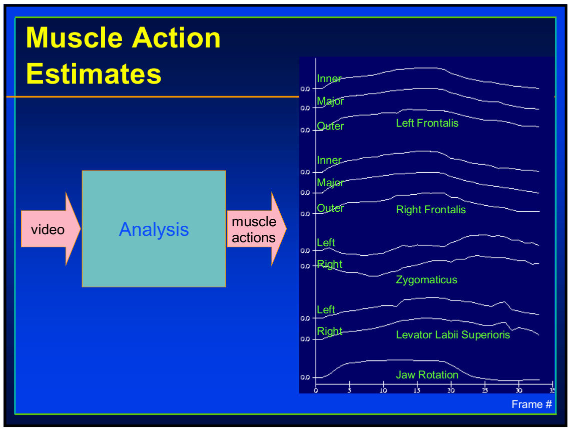
Fiducial Points

Muscle action estimation

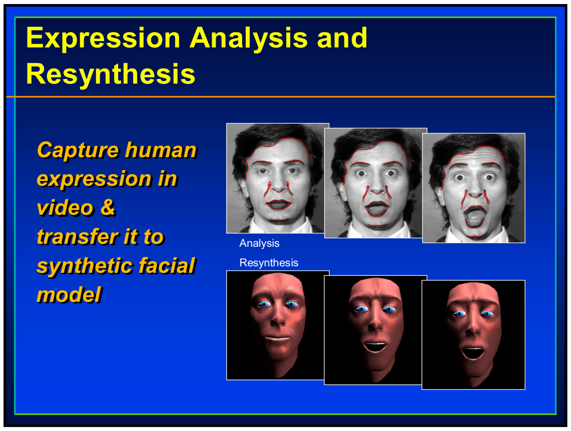
- Calibrate fiducials on neutral face
- Track facial features
- Decode fiducial displacements

Neutral Pose

Max Surprise Pose



Muscle Action Estimates



Expression Analysis and Resynthesis

Capture human expression in video & transfer it to synthetic facial model

Analysis

Resynthesis

NN Estimation of Muscle Actions

Trained neural network
muscle control

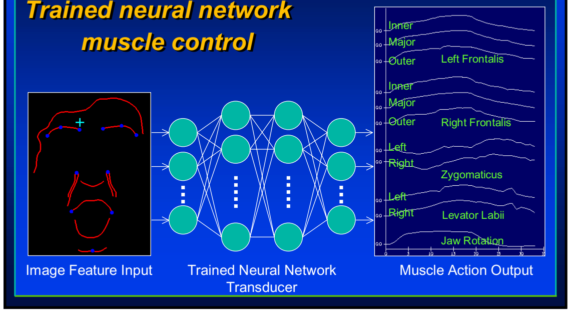
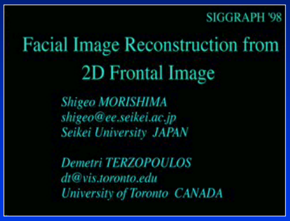


Image Feature Input

Trained Neural Network Transducer

Muscle Action Output

Expression Analysis and Resynthesis



SIGGRAPH '98

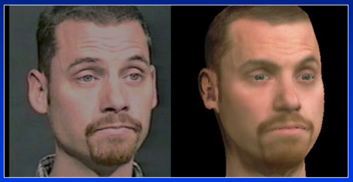
Facial Image Reconstruction from
2D Frontal Image

Shigeo MORISHIMA
shigeo@ee.seikei.ac.jp
Seikei University JAPAN

Demetri TERZOPOULOS
dt@vis.toronto.edu
University of Toronto CANADA

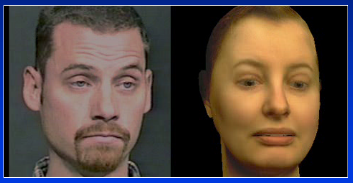
Performance-Based Animation

[Pighin, Szeliski, Salezin, 1999]



Performance-Based Animation

[Pighin, Szeliski, Salezin, 1999]



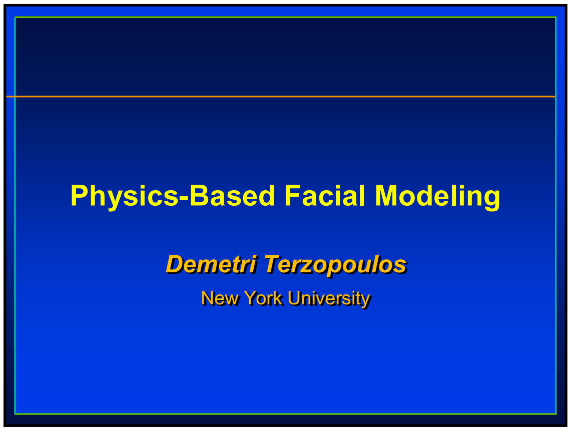
Acknowledgements

Collaborators:

- Yuencheng Lee — UofT (realistic facial model)
- Keith Waters — LifeFX, Inc., Boston
- Shigeo Morishima — Seikei University, Tokyo

Additional info:

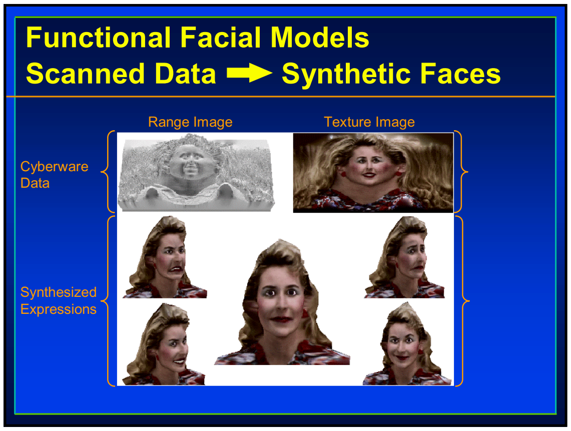
www.mrl.nyu.edu/~dt



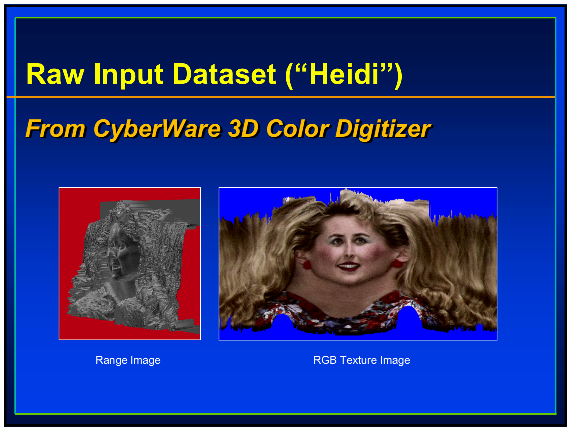
Physics-Based Facial Modeling

Demetri Terzopoulos

New York University



Functional Facial Models
Scanned Data ➡ Synthetic Faces

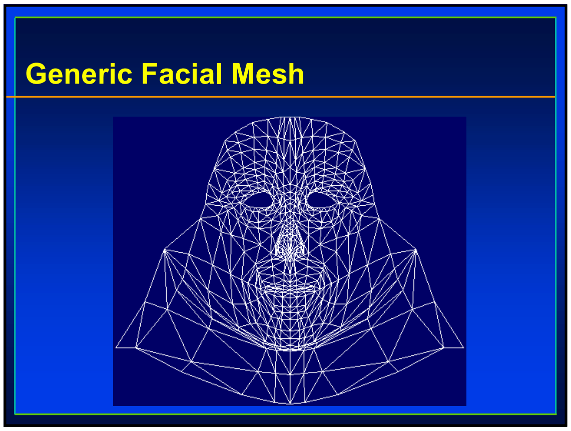


Raw Input Dataset ("Heidi")

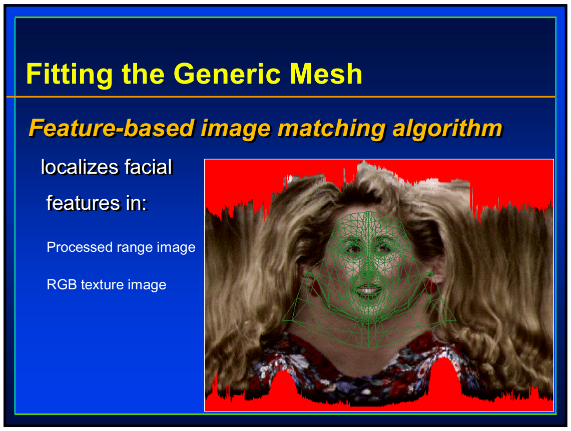
From CyberWare 3D Color Digitizer

Range Image

RGB Texture Image



Generic Facial Mesh



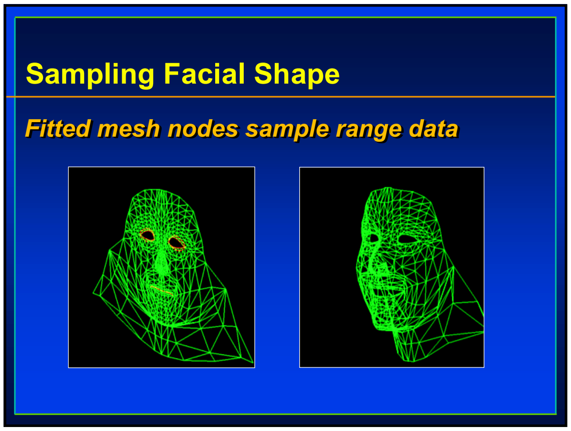
Fitting the Generic Mesh

Feature-based image matching algorithm

localizes facial features in:

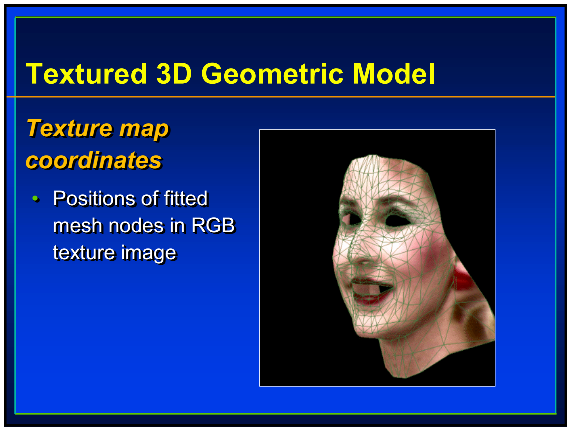
Processed range image

RGB texture image



Sampling Facial Shape

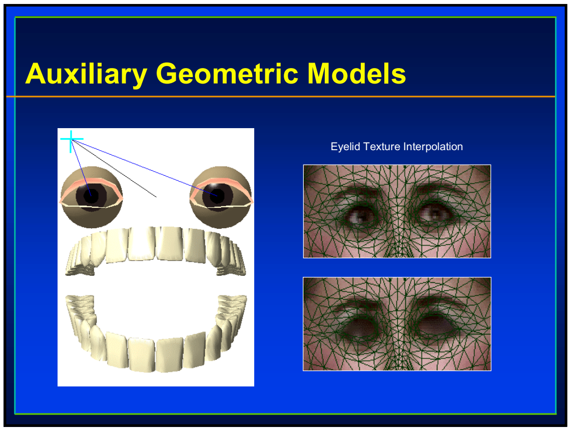
Fitted mesh nodes sample range data



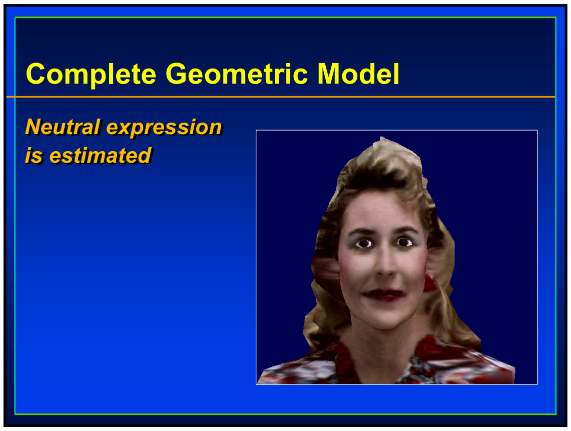
Textured 3D Geometric Model

Texture map coordinates

- Positions of fitted mesh nodes in RGB texture image

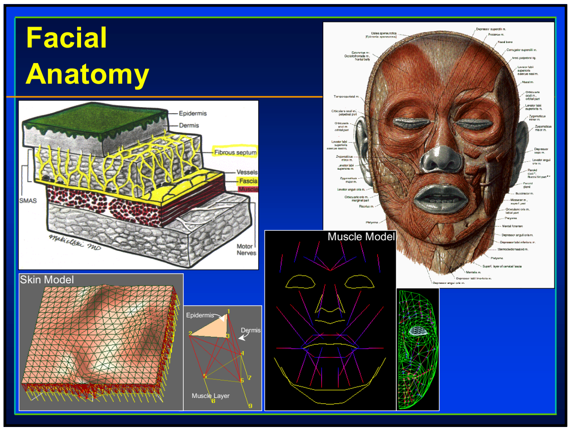


Auxiliary Geometric Models

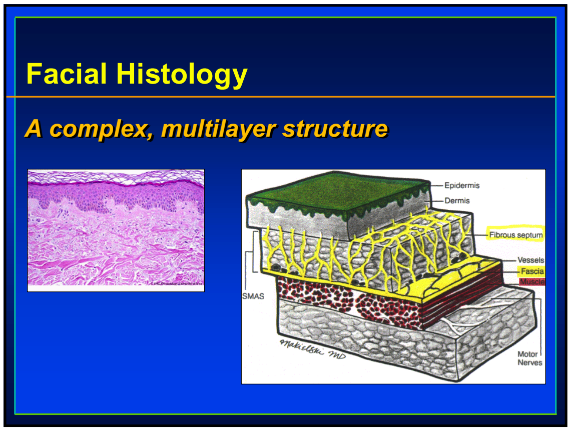


Complete Geometric Model

Neutral expression is estimated

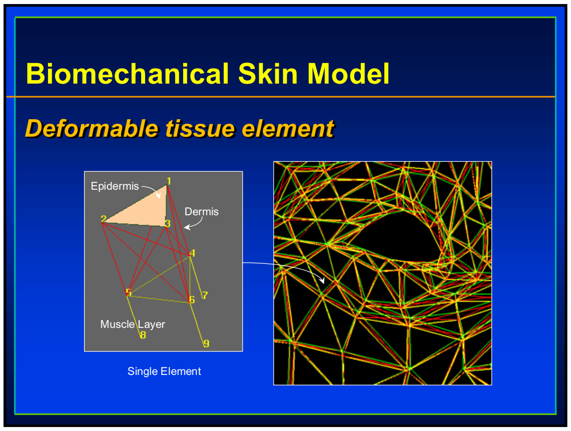


Facial Anatomy



Facial Histology

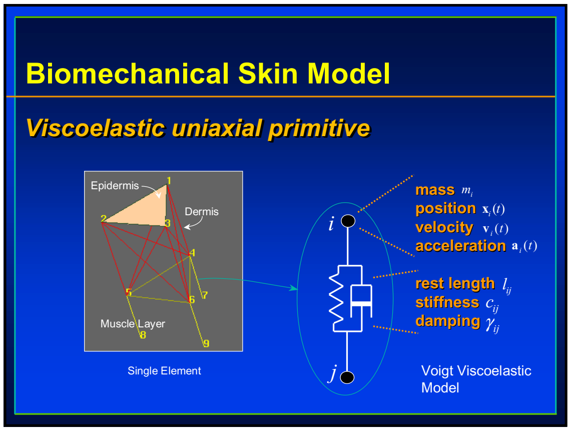
A complex, multilayer structure



Biomechanical Skin Model

Deformable tissue element

Single Element



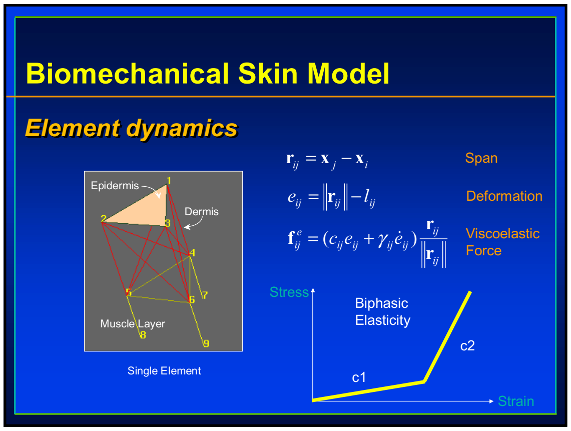
Biomechanical Skin Model

Viscoelastic uniaxial primitive

mass m_i
position $\mathbf{x}_i(t)$
velocity $\mathbf{v}_i(t)$
acceleration $\mathbf{a}_i(t)$

rest length l_{ij}
stiffness c_{ij}
damping γ_{ij}

Voigt Viscoelastic Model



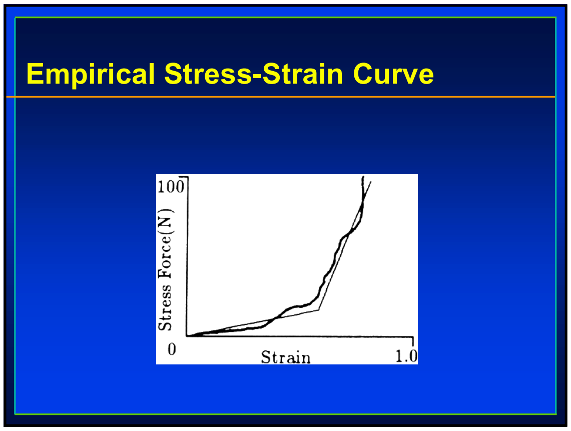
Biomechanical Skin Model

Element dynamics

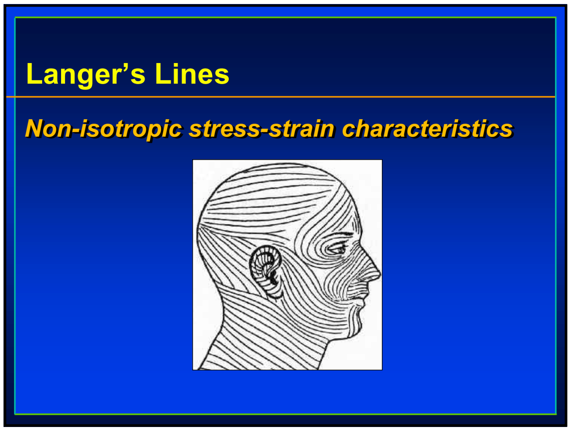
$\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ Span

$e_{ij} = \|\mathbf{r}_{ij}\| - l_{ij}$ Deformation

$\mathbf{f}^e_{ij} = (c_{ij}e_{ij} + \gamma_{ij}\dot{e}_{ij})\frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|}$ Viscoelastic Force

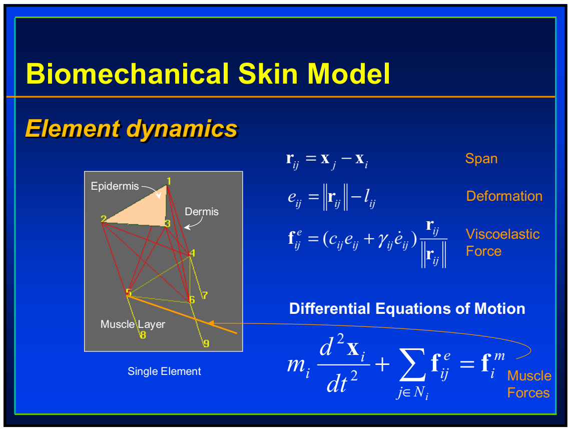


Empirical Stress-Strain Curve



Langer's Lines

Non-isotropic stress-strain characteristics



Biomechanical Skin Model

Element dynamics

$\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ Span

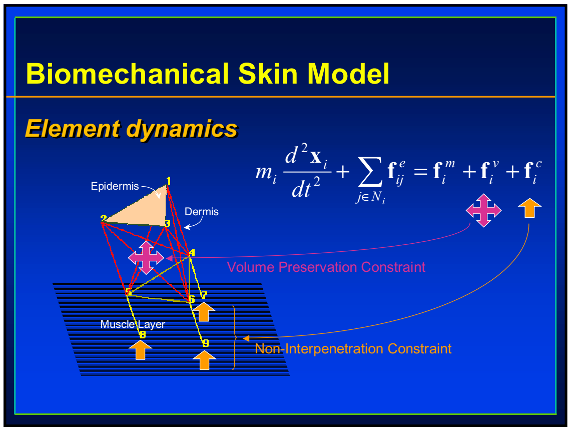
$e_{ij} = \|\mathbf{r}_{ij}\| - l_{ij}$ Deformation

$\mathbf{f}^e_{ij} = (c_{ij}e_{ij} + \gamma_{ij}\dot{e}_{ij})\frac{\mathbf{r}_{ij}}{\|\mathbf{r}_{ij}\|}$ Viscoelastic Force

Differential Equations of Motion

$$m_i \frac{d^2\mathbf{x}_i}{dt^2} + \sum_{j\in N_i} \mathbf{f}^e_{ij} = \mathbf{f}^m_i$$

Muscle Forces



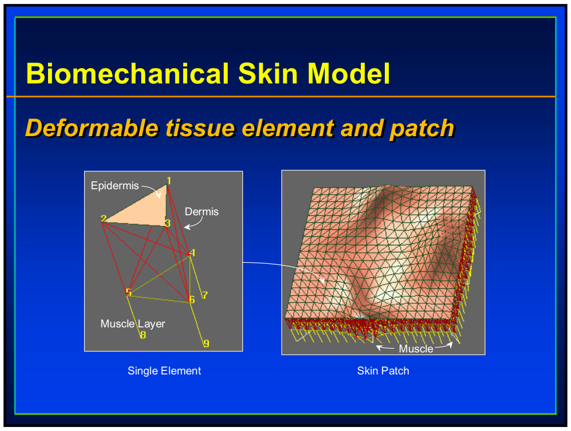
Biomechanical Skin Model

Element dynamics

$$m_i \frac{d^2\mathbf{x}_i}{dt^2} + \sum_{j\in N_i} \mathbf{f}^e_{ij} = \mathbf{f}^m_i + \mathbf{f}^v_i + \mathbf{f}^c_i$$

Volume Preservation Constraint

Non-Interpenetration Constraint

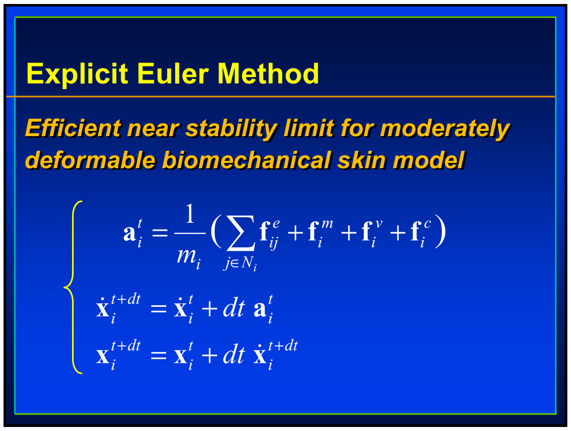


Biomechanical Skin Model

Deformable tissue element and patch

Single Element

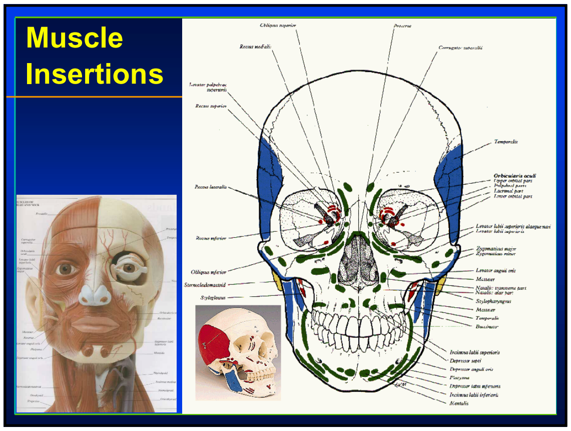
Skin Patch



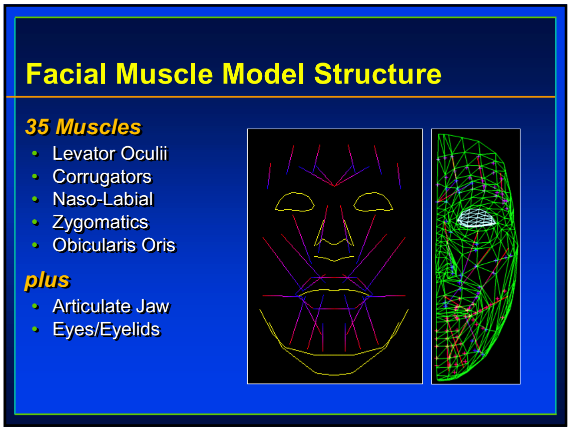
Explicit Euler Method

Efficient near stability limit for moderately deformable biomechanical skin model

$$\begin{cases} \mathbf{a}_i^t = \dfrac{1}{m_i}\left(\sum_{j\in N_i}\mathbf{f}_{ij}^e + \mathbf{f}_i^m + \mathbf{f}_i^v + \mathbf{f}_i^c\right) \\ \dot{\mathbf{x}}_i^{t+dt} = \dot{\mathbf{x}}_i^t + dt\,\mathbf{a}_i^t \\ \mathbf{x}_i^{t+dt} = \mathbf{x}_i^t + dt\,\dot{\mathbf{x}}_i^{t+dt} \end{cases}$$



Muscle Insertions



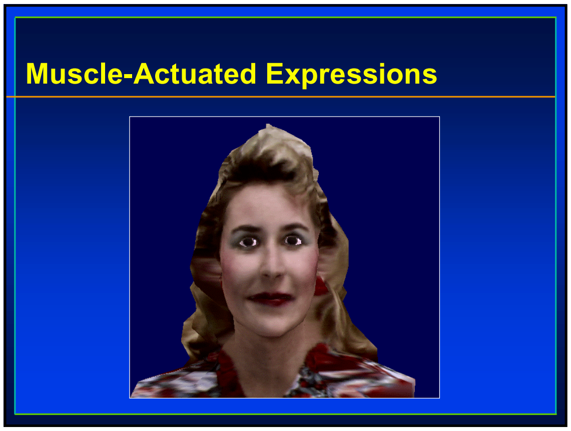
Facial Muscle Model Structure

35 Muscles

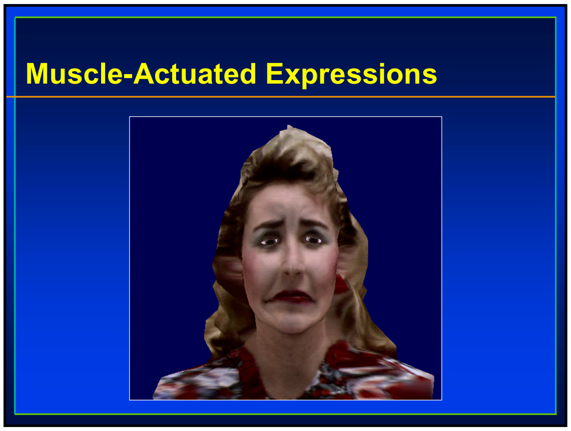
- Levator Oculii
- Corrugators
- Naso-Labial
- Zygomatics
- Obicularis Oris

plus

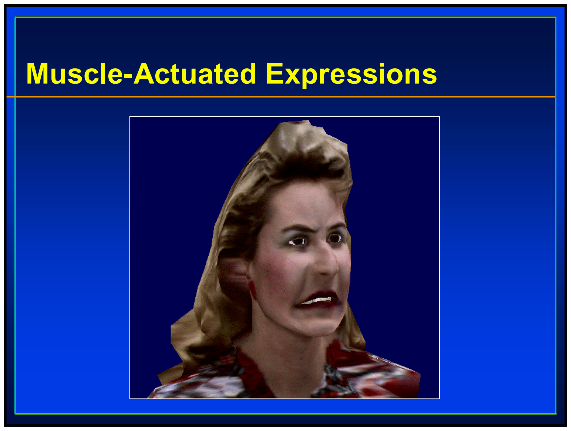
- Articulate Jaw
- Eyes/Eyelids



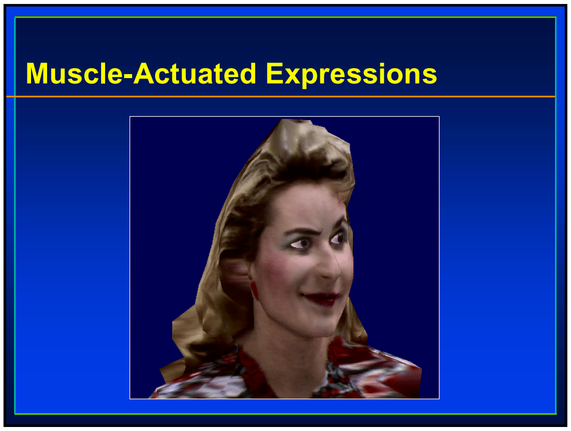
Muscle-Actuated Expressions



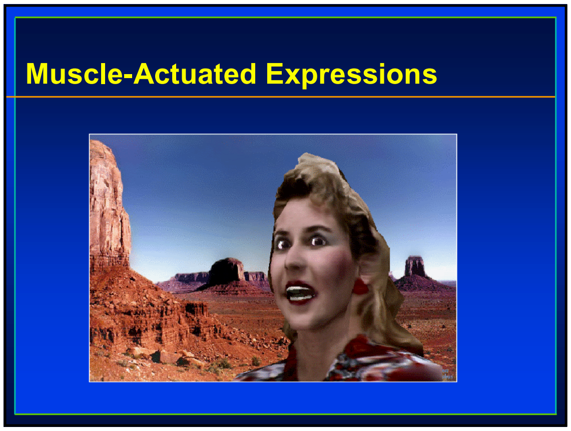
Muscle-Actuated Expressions



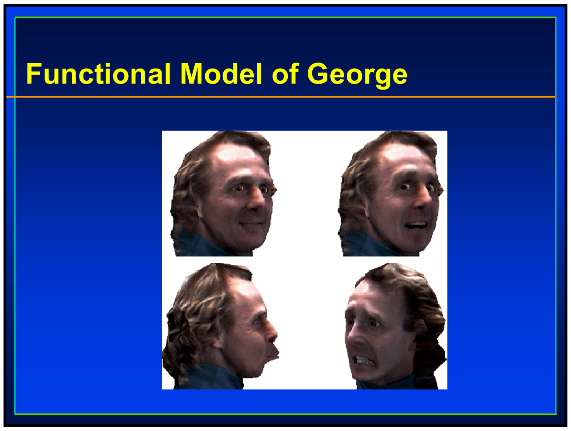
Muscle-Actuated Expressions



Muscle-Actuated Expressions

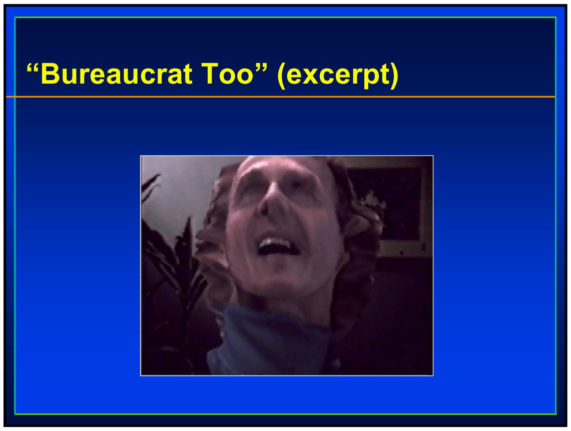


Muscle-Actuated Expressions

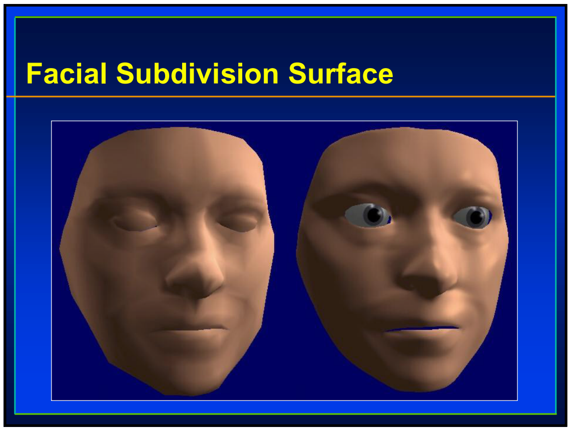


Functional Model of George

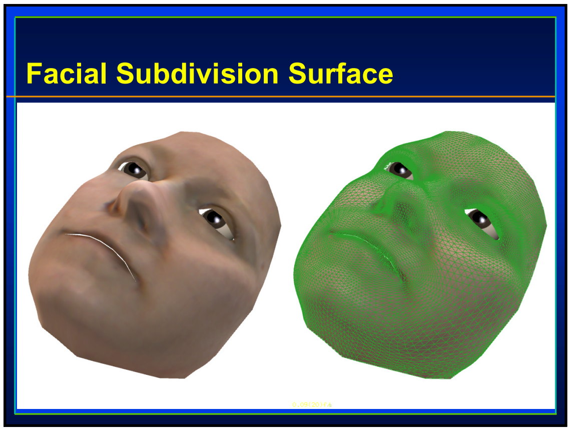
George in “Bureaucrat Too”



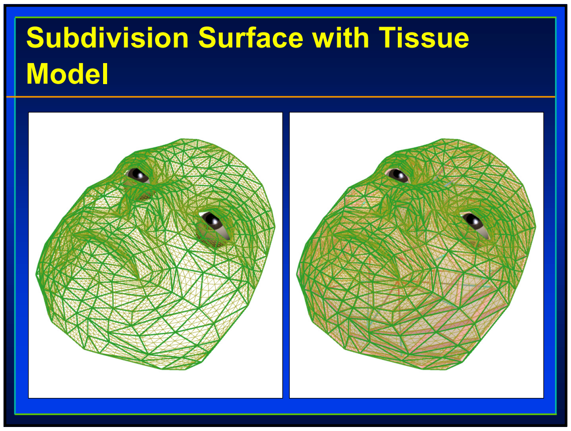
“Bureaucrat Too” (excerpt)



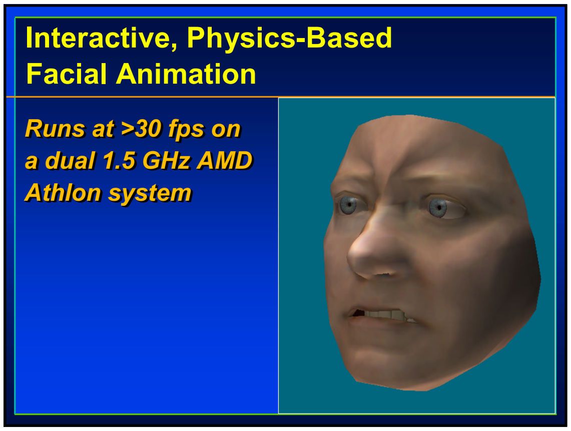
Facial Subdivision Surface



Facial Subdivision Surface



Subdivision Surface with Tissue Model



Interactive, Physics-Based Facial Animation

Runs at >30 fps on a dual 1.5 GHz AMD Athlon system

Realistic Modeling for Facial Animation

Yuencheng Lee[1], Demetri Terzopoulos[1], and Keith Waters[2]
University of Toronto[1] and Digital Equipment Corporation[2]

Abstract

A major unsolved problem in computer graphics is the construction and animation of realistic human facial models. Traditionally, facial models have been built painstakingly by manual digitization and animated by ad hoc parametrically controlled facial mesh deformations or kinematic approximation of muscle actions. Fortunately, animators are now able to digitize facial geometries through the use of scanning range sensors and animate them through the dynamic simulation of facial tissues and muscles. However, these techniques require considerable user input to construct facial models of individuals suitable for animation. In this paper, we present a methodology for automating this challenging task. Starting with a structured facial mesh, we develop algorithms that automatically construct functional models of the heads of human subjects from laser-scanned range and reflectance data. These algorithms automatically insert contractile muscles at anatomically correct positions within a dynamic skin model and root them in an estimated skull structure with a hinged jaw. They also synthesize functional eyes, eyelids, teeth, and a neck and fit them to the final model. The constructed face may be animated via muscle actuations. In this way, we create the most authentic and functional facial models of individuals available to date and demonstrate their use in facial animation.

CR Categories: I.3.5 **[Computer Graphics]:** Physically based modeling; I.3.7 **[Computer Graphics]:** Animation.

Additional Keywords: Physics-based Facial Modeling, Facial Animation, RGB/Range Scanners, Feature-Based Facial Adaptation, Texture Mapping, Discrete Deformable Models.

1 Introduction

Two decades have passed since Parke's pioneering work in animating faces [13]. In the span of time, significant effort has been devoted to the development of computational models of the human face for applications in such diverse areas as entertainment, low bandwidth teleconferencing, surgical facial planning, and virtual reality. However, the task of accurately modeling the expressive human face by computer remains a major challenge.

Traditionally, computer facial animation follows three basic procedures: (1) design a 3D facial mesh, (2) digitize the 3D mesh, and (3) animate the 3D mesh in a controlled fashion to simulate facial actions.

In procedure (1), it is desirable to have a refined topological mesh that captures the facial geometry. Often this entails digitizing as many nodes as possible. Care must be taken not to oversample the surface because there is a trade-off between the number of nodes and the computational cost of the model. Consequently, meshes developed to date capture the salient features of the face with as few nodes as possible (see [17, 14, 21, 9, 23] for several different mesh designs).

In procedure (2), a general 3D digitization technique uses photogrammetry of several images of the face taken from different angles. A common technique is to place markers on the face that can be seen from two or more cameras. An alternative technique is to manually digitize a plaster cast of the face using manual 3D digitization devices such as orthogonal magnetic fields sound captors [9], or one to two photographs [9, 7, 1]. More recently, automated laser range finders can digitize on the order of 10^5 3D points from a solid object such as a person's head and shoulders in just a few seconds [23].

In procedure (3), an animator must decide which mesh nodes to articulate and how much they should be displaced in order to produce a specific facial expression. Various approaches have been proposed for deforming a facial mesh to produce facial expressions; for example, parameterized models [14, 15], control-point models [12, 7], kinematic muscle models [21, 9], a texture-map-assembly model [25], a spline model [11], feature-tracking models [24, 16], a finite element model [6], and dynamic muscle models [17, 20, 8, 3].

1.1 Our Approach

The goal of our work is to automate the challenging task of creating realistic facial models of individuals suitable for animation. We develop an algorithm that begins with cylindrical range and reflectance data acquired by a Cyberware scanner and automatically constructs an efficient and fully functional model of the subject's head, as shown in Plate 1. The algorithm is applicable to various individuals (Plate 2 shows the raw scans of several individuals). It proceeds in two steps:

In step 1, the algorithm adapts a well-structured face mesh from [21] to the range and reflectance data acquired by scanning the subject, thereby capturing the shape of the subject's face. This approach has significant advantages because it avoids repeated manual modification of control parameters to compensate for geometric variations in the facial features from person to person. More specifically, it allows the automatic placement of facial muscles and enables the use of a single control process across different facial models.

The generic face mesh is adapted automatically through an image analysis technique that searches for salient local minima and maxima in the range image of the subject. The search is directed according to the known relative positions of the nose, eyes, chin, ears, and other facial features with respect to the generic mesh. Facial muscle emergence and attachment points are also known relative to the generic mesh and are adapted automatically as the mesh is conformed to the scanned data.

In step 2, the algorithm elaborates the geometric model constructed in step 1 into a functional, physics-based model of the subject's face which is capable of facial expression, as shown in the lower portion of Plate 1.

We follow the physics-based facial modeling approach proposed

[1] Department of Computer Science, 10 King's College Road, Toronto, ON, Canada, M5S 1A4. {vlee | dt}@cs.toronto.edu

[2] Cambridge Research Lab., One Kendall Square, Cambridge, MA 02139. waters@crl.dec.com

Published in the Proceedings of SIGGRAPH 95 (Los Angeles, CA, August, 1995). In *Computer Graphics* Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 55–62.

by Terzopoulos and Waters [20]. Its basic features are that it animates facial expressions by contracting synthetic muscles embedded in an anatomically motivated model of skin composed of three spring-mass layers. The physical simulation propagates the muscle forces through the physics-based synthetic skin thereby deforming the skin to produce facial expressions. Among the advantages of the physics-based approach are that it greatly enhances the degree of realism over purely geometric facial modeling approaches, while reducing the amount of work that must be done by the animator. It can be computationally efficient. It is also amenable to improvement, with an increase in computational expense, through the use of more sophisticated biomechanical models and more accurate numerical simulation methods.

We propose a more accurate biomechanical model for facial animation compared to previous models. We develop a new biomechanical facial skin model which is simpler and better than the one proposed in [20]. Furthermore, we argue that the skull is an important biomechanical structure with regard to facial expression [22]. To date, the skin-skull interface has been underemphasized in facial animation despite its importance in the vicinity of the articulate jaw; therefore we improve upon previous facial models by developing an algorithm to estimate the skull structure from the acquired range data, and prevent the synthesized facial skin from penetrating the skull.

Finally, our algorithm includes an articulated neck and synthesizes subsidiary organs, including eyes, eyelids, and teeth, which cannot be adequately imaged or resolved in the scanned data, but which are nonetheless crucial for realistic facial animation.

2 Generic Face Mesh and Mesh Adaptation

The first step of our approach to constructing functional facial models of individuals is to scan a subject using a Cyberware Color Digitizer$^{\text{TM}}$. The scanner rotates 360 degrees around the subject, who sits motionless on a stool as a laser stripe is projected onto the head and shoulders. Once the scan is complete, the device has acquired two registered images of the subject: a range image (Figure 1) — a topographic map that records the distance from the sensor to points on the facial surface, and a reflectance (RGB) image (Figure 2) — which registers the color of the surface at those points. The images are in cylindrical coordinates, with longitude (0–360) degrees along the x axis and vertical height along the y axis. The resolution of the images is typically 512 × 256 pixels (cf. Plate 1)

The remainder of this section describes an algorithm which reduces the acquired geometric and photometric data to an efficient geometric model of the subject's head. The algorithm is a two-part process which repairs defects in the acquired images and conforms a generic facial mesh to the processed images using a feature-based matching scheme. The resulting mesh captures the facial geometry as a polygonal surface that can be texture mapped with the full resolution reflectance image, thereby maintaining a realistic facsimile of the subject's face.

2.1 Image Processing

One of the problems of range data digitization is illustrated in Figure 1(a). In the hair area, in the chin area, nostril area, and even in the pupils, laser beams tend to disperse and the sensor observes no range value for these corresponding 3D surface points. We must correct for missing range and texture information.

We use a *relaxation method* to interpolate the range data. In particular, we apply a membrane interpolation method described in [18]. The relaxation interpolates values for the missing points so as to bring them into successively closer agreement with surrounding points by repeatedly indexing nearest neighbor values. Intuitively, it stretches an elastic membrane over the gaps in the surface. The images interpolated through relaxation are shown in Figure 1(b) and illustrate improvements in the hair area and chin area. Relaxation works effectively when the range surface is smooth, and particularly in the case of human head range data, the smoothness requirement of the solutions is satisfied quite effectively.

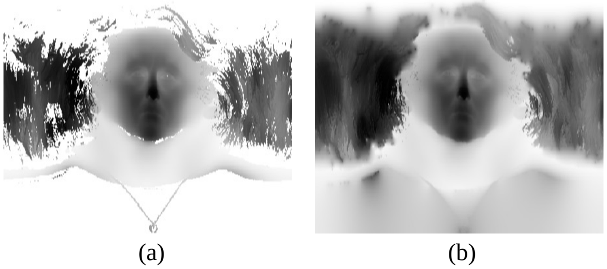


Figure 1: (a) Range data of "Grace" from a Cyberware scanner. (b) Recovered plain data.

Figure 2(a) shows two 512 × 256 reflectance (RGB) texture maps as monochrome images. Each reflectance value represents the surface color of the object in cylindrical coordinates with corresponding longitude (0–360 degrees) and latitude. Like range images, the acquired reflectance images are lacking color information at certain points. This situation is especially obvious in the hair area and the shoulder area (see Figure 2(a)). We employ the membrane relaxation approach to interpolate the texture image by repeated averaging of neighboring known colors. The original texture image in Figure 2(a) can be compared with the interpolated texture image in Figure 2(b).

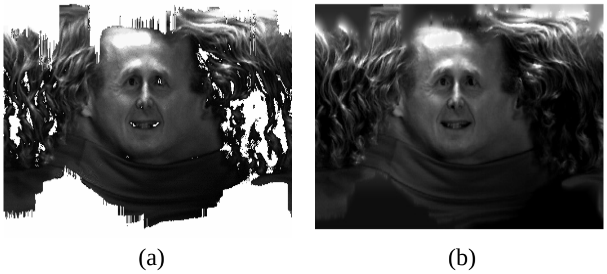


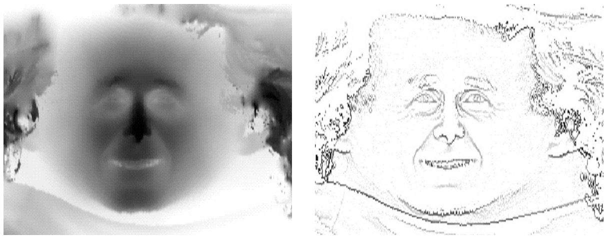
Figure 2: (a) Texture data of "George" with void points displayed in white and (b) texture image interpolated using relaxation method.

The method is somewhat problematic in the hair area where range variations may be large and there is a relatively high percentage of missing surface points. A thin-plate relaxation algorithm [18] may be more effective in these regions because it would fill in the larger gaps with less "flattening" than a membrane [10].

Although the head structure in the cylindrical laser range data is distorted along the longitudinal direction, important features such as the slope changes of the nose, forehead, chin, and the contours of the mouth, eyes, and nose are still discernible. In order to locate the contours of those facial features for use in adaptation (see below), we use a modified Laplacian operator (applied to the discrete image through local pixel differencing) to detect edges from the range map shown in Figure 3(a) and produce the field function in Fig. 3(b). For details about the operator, see [8]. The field function highlights important features of interest. For example, the local maxima of the modified Laplacian reveals the boundaries of the lips, eyes, and chin.

2.2 Generic Face Mesh and Mesh Adaptation

The next step is to reduce the large arrays of data acquired by the scanner into a parsimonious geometric model of the face that can eventually be animated efficiently. Motivated by the adaptive meshing techniques [19] that were employed in [23], we significantly



(a) (b)

Figure 3: (a) Original range map. (b) Modified Laplacian field function of (a).

improved the technique by adapting a generic face mesh to the data. Figure 4 shows the planar generic mesh which we obtain through a cylindrical projection of the 3D face mesh from [21]. One of the advantages of the generic mesh is that it has well-defined features which form the basis for accurate feature based adaptation to the scanned data and automatic scaling and positioning of facial muscles as the mesh is deformed to fit the images. Another advantage is that it automatically produces an efficient triangulation, with finer triangles over the highly curved and/or highly articulate regions of the face, such as the eyes and mouth, and larger triangles elsewhere.

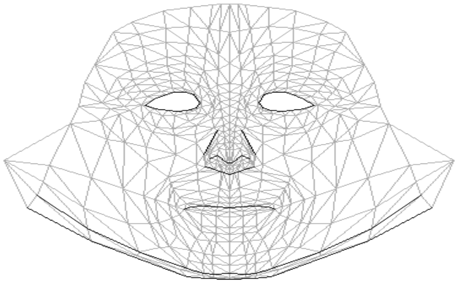


Figure 4: Facial portion of generic mesh in 2D cylindrical coordinates. Dark lines are features for adaptation.

We label all facial feature nodes in the generic face prior to the adaptation step. The feature nodes include eye contours, nose contours, mouth contours, and chin contours.

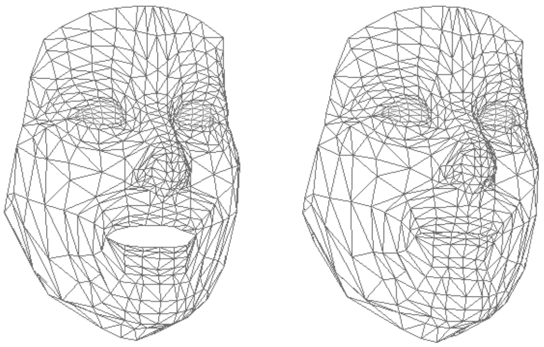
For any specific range image and its positive Laplacian field function (Figure 3), the generic mesh adaptation procedure performs the following steps to locate feature points in the range data (see [8] for details):

Mesh Adaptation Procedures

1. Locate nose tip
2. Locate chin tip
3. Locate mouth contour
4. Locate chin contour
5. Locate ears
6. Locate eyes
7. Activate spring forces
8. Adapt hair mesh
9. Adapt body mesh
10. Store texture coordinates

Once the mesh has been fitted by the above feature based matching technique (see Plate 3), the algorithm samples the range image at the location of the nodes of the face mesh to capture the facial geometry, as is illustrated in Figure 5.

The node positions also provide texture map coordinates that are used to map the full resolution color image onto the triangles (see Plate 3).



(a) (b)

Figure 5: (a) Generic geometric model conformed to Cyberware scan of "Heidi". (b) Same as (a). Note that "Heidi's" mouth is now closed, subsequent to estimation of the relaxed face geometry.

2.3 Estimation of Relaxed Face Model

Ideally, the subject's face should be in a neutral, relaxed expression when he or she is being scanned. However, the scanned woman in the "Heidi" dataset is smiling and her mouth is open (see Plate 2). We have made our algorithm tolerant of these situations. To construct a functional model, it is important to first estimate the relaxed geometry. That is, we must infer what the "Heidi" subject would look like had her face been in a relaxed pose while she was being scanned. We therefore estimate the range values of the closed mouth contour from the range values of the open mouth contour by the following steps:

1. Perform adaptation procedures in Sec. 2.2 without step 3.
2. Store nodal longitude/latitude into adapted face model.
3. Perform lip adaptation in step 3 in sec. 2.2
4. Store nodal range values into adapted face model.

As a result, the final reconstructed face model in Figure 5(b) will have a relaxed mouth because the longitude and latitude recorded is the default shape of our closed mouth model (see Figure 4). Moreover, the shape of the final reconstructed face is still faithful to the head data because the range value at each facial nodal point is obtained correctly after the lip adaptation procedure has been performed. Relaxing the face shown in Figure 5(a) results in the image in Figure 5(b) (with eyelids inserted — see below).

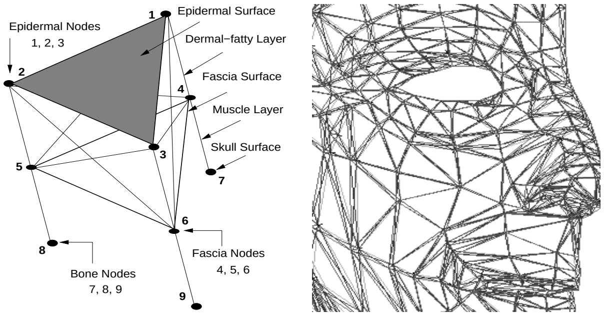
3 The Dynamic Skin and Muscle Model

This section describes how our system proceeds with the construction of a fully functional model of the subject's face from the facial mesh produced by the adaptation algorithm described in the previous section. To this end, we automatically create a dynamic model of facial tissue, estimate a skull surface, and insert the major muscles of facial expression into the model. The following sections describe each of these components. We also describe our high-performance parallel, numerical simulation of the dynamic facial tissue model.

3.1 Layered Synthetic Tissue Model

The skull is covered by deformable tissue which has five distinct layers [4]. Four layers—epidermis, dermis, sub-cutaneous connective tissue, and fascia—comprise the skin, and the fifth consists of the muscles of facial expression. Following [20], and in accordance with the structure of real skin [5], we have designed a new, synthetic tissue model (Figure 6(a)).

The tissue model is composed of triangular prism elements (see Figure 6(a)) which match the triangles in the adapted facial mesh. The epidermal surface is defined by nodes 1, 2, and 3, which are connected by epidermal springs. The epidermis nodes are also connected by dermal-fatty layer springs to nodes 4, 5, and 6, which define the fascia surface. Fascia nodes are interconnected by fascia



(a) (b)

Figure 6: (a) Triangular skin tissue prism element. (b) Close-up view of right side of an individual with conformed elements.

springs. They are also connected by muscle layer springs to skull surface nodes 7, 8, 9.

Figure 9(b) shows 684 such skin elements assembled into an extended skin patch. Several synthetic muscles are embedded into the muscle layer of the skin patch and the figure shows the skin deformation due to muscle contraction. Muscles are fixed in an estimated bony subsurface at their point of emergence and are attached to fascia nodes as they run through several tissue elements. Figure 6(b) shows a close-up view of the right half of the facial tissue model adapted to an individual's face which consists of 432 elements.

3.2 Discrete Deformable Models (DDMs)

A discrete deformable model has a node-spring-node structure, which is a uniaxial finite element. The data structure for the node consists of the nodal mass m_i, position $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]'$, velocity $\mathbf{v}_i = d\mathbf{x}_i/dt$, acceleration $\mathbf{a}_i = d^2\mathbf{x}_i/dt^2$, and net nodal forces $\mathbf{f}_i^{\mathrm{n}}(t)$. The data structure for the spring in this DDM consists of pointers to the head node i and the tail node j which the spring interconnects, the natural or rest length l_k of the spring, and the spring stiffness c_k.

3.3 Tissue Model Spring Forces

By assembling the discrete deformable model according to histological knowledge of skin (see Figure 6(a)), we are able to construct an anatomically consistent, albeit simplified, tissue model. Figure 6(b) shows a close-up view of the tissue model around its eye and nose parts of a face which is automatically assembled by following the above approach.

- The force spring j exerts on node i is

  $\mathbf{g}_j = c_j(l_j - l_j^r)\mathbf{s}_j$

  – each layer has its own stress-strain relationship c_j and the dermal-fatty layer uses biphasic springs (non-constant c_j) [20]

  – l_j^r and $l_j = ||\mathbf{x}_j - \mathbf{x}_i||$ are the rest and current lengths for spring j

  – $\mathbf{s}_j = (\mathbf{x}_j - \mathbf{x}_i)/l_j$ is the spring direction vector for spring j

3.4 Linear Muscle Forces

The muscles of facial expression, or the muscular plate, spreads out below the facial tissue. The facial musculature is attached to the skin tissue by short elastic tendons at many places in the fascia, but is fixed to the facial skeleton only at a few points. Contractions of the facial muscles cause movement of the facial tissue. We model 28 of the primary facial muscles, including the zygomatic major and minor, frontalis, nasii, corrugator, mentalis, buccinator, and angulii depressor groups. Plate 4 illustrates the effects of automatic scaling and positioning of facial muscle vectors as the generic mesh adapts to different faces.

To better emulate the facial muscle attachments to the fascia layer in our model, a group of fascia nodes situated along the muscle path—i.e., within a predetermined distance from a central muscle vector, in accordance with the muscle width—experience forces from the contraction of the muscle. The face construction algorithm determines the nodes affected by each muscle in a precomputation step.

To apply muscle forces to the fascia nodes, we calculate a force for each node by multiplying the muscle vector with a force length scaling factor and a force width scaling factor (see Figure 7(a)). Function Θ_1 (Figure 8(a)) scales the muscle force according to the length ratio $\varepsilon_{j,i}$, while Θ_2 (Figure 8(b)) scales it according to the width $\omega_{j,i}$ at node i of muscle j:

$$\begin{aligned}\varepsilon_{j,i} &= ((\mathbf{m}_j^F - \mathbf{x}_i)\cdot\mathbf{m}_j)/(\|\mathbf{m}_j^A - \mathbf{m}_j^F\|)\\ \omega_{j,i} &= \|\mathbf{p}_i - (\mathbf{p}_i\cdot\mathbf{n}_j)\mathbf{n}_j\|\end{aligned}$$

- The force muscle j exerts on node i is

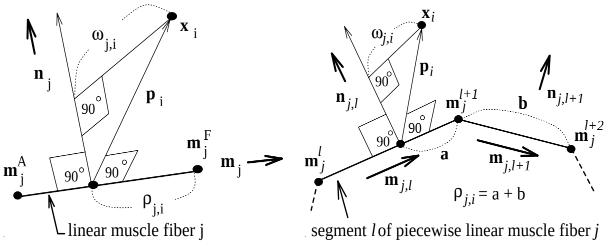
  $\mathbf{f}_i^j = \Theta_1(\varepsilon_{j,i})\Theta_2(\omega_{j,i})\mathbf{m}_j$

  – Θ_1 scales the force according to the distance ratio $\varepsilon_{j,i}$, where $\varepsilon_{j,i} = \rho_{j,i}/d_j$, with d_j the muscle j length.

  – Θ_2 scales the force according to the width ratio $\omega_{j,i}/w_j$, with w_j the muscle j width.

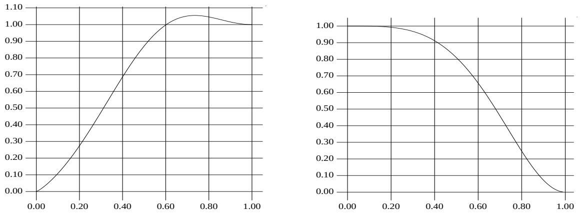
  – $\mathbf{m}_j$ is the normalized muscle vector for muscle j

Note that the muscle force is scaled to zero at the root of the muscle fiber in the bone and reaches its full strength near the end of the muscle fiber. Figure 9(b) shows an example of the effect of muscle forces applied to a synthetic skin patch.



(a) (b)

Figure 7: (a) Linear muscle fiber. (b) Piecewise linear muscle fiber.



(a) (b)

Figure 8: (a) Muscle force scaling function Θ_1 wrt $\varepsilon_{j,i}$, (b) Muscle force scaling function Θ_2 wrt $\omega_{j,i}/w_j$

3.5 Piecewise Linear Muscle Forces

In addition to using linear muscle fibers in section 3.4 to simulate sheet facial muscles like the frontalis and the zygomatics, we also model sphincter muscles, such as the orbicularis oris circling the mouth, by generalizing the linear muscle fibers to be piecewise linear and allowing them to attach to fascia at each end of the segments. Figure 7(b) illustrates two segments of an N-segment piecewise linear muscle j showing three nodes $\mathbf{m}_j^l$, $\mathbf{m}_j^{l+1}$, and $\mathbf{m}_j^{l+2}$. The unit vectors $\mathbf{m}_{j,l}$, $\mathbf{m}_{j,l+1}$ and $\mathbf{n}_{j,l}$, $\mathbf{n}_{j,l+1}$ are parallel and normal to the segments, respectively. The figure indicates fascia node i at $\mathbf{x}_i$, as well as the distance $\rho_{j,i} = a + b$, the width $\omega_{j,i}$, and the perpendicular vector $\mathbf{p}_i$ from fascia node i to the nearest segment of the muscle. The length ratio $\varepsilon_{j,i}$ for fascia node i in muscle fiber j is

$$\varepsilon_{j,i} = \frac{(\mathbf{m}_j^{l+1} - \mathbf{x}_i) \cdot \mathbf{m}_{j,l} + \sum_{k=l+1}^{N} \parallel \mathbf{m}_j^{k+1} - \mathbf{m}_j^k \parallel}{\sum_{k=1}^{N} \parallel \mathbf{m}_j^{k+1} - \mathbf{m}_j^k \parallel}$$

The width $\omega_{j,i}$ calculation is the same as for linear muscles. The remaining muscle force computations are the same as in section 3.4. Plate 4 shows all the linear muscles and the piecewise linear sphincter muscles around the mouth.

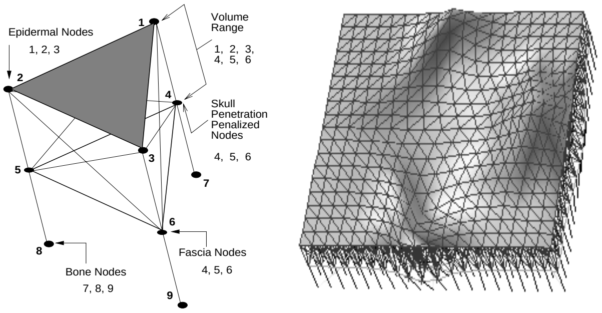
3.6 Volume Preservation Forces

In order to faithfully exhibit the incompressibility [2] of real human skin in our model, a volume constraint force based on the change of volume (see Figure 9(a)) and displacements of nodes is calculated and applied to nodes. In Figure 9(b) the expected effect of volume preservation is demonstrated. For example, near the origin of the muscle fiber, the epidermal skin is bulging out, and near the end of the muscle fiber, the epidermal skin is depressed.

- The volume preservation force element e exerts on nodes i in element e is

  $$\mathbf{q}_i^e = k_1(V^e - \tilde{V}^e)\mathbf{n}_i^e + k_2(\mathbf{p}_i^e - \tilde{\mathbf{p}}_i^e)$$

  - $\tilde{V}^e$ and V^e are the rest and current volumes for e
  - $\mathbf{n}_i^e$ is the epidermal normal for epidermal node i
  - $\tilde{\mathbf{p}}_i^e$ and $\mathbf{p}_i^e$ are the rest and current nodal coordinates for node i with respect to the center of mass of e
  - k_1, k_2 are force scaling constants



(a) (b)

Figure 9: (a) Volume preservation and skull nonpenetration element. (b) Assembled layered tissue elements under multiple muscle forces.

3.7 Skull Penetration Constraint Forces

Because of the underlying impenetrable skull of a human head, the facial tissue during a facial expression will slide over the underlying bony structure. With this in mind, for each individual's face model reconstructed from the laser range data, we estimate the skull surface normals to be the surface normals in the range data image. The skull is then computed as an offset surface. To prevent nodes from penetrating the estimated skull (see Figure 9(a)), we apply a skull non-penetration constraint to cancel out the force component on the fascia node which points into the skull; therefore, the resulting force will make the nodes slide over the skull.

- The force to penalize fascia node i during motion is:

  $$\mathbf{s}_i = \begin{cases} -(\mathbf{f}_i^n \cdot \mathbf{n}_i)\mathbf{n}_i & \text{when } \mathbf{f}_i^n \cdot \mathbf{n}_i < 0 \\ \mathbf{0} & \text{otherwise} \end{cases}$$

  - $\mathbf{f}_i^n$ is the net force on fascia node i
  - $\mathbf{n}_i$ is the nodal normal of node i

3.8 Equations of Motion for Tissue Model

Newton's law of motion governs the response of the tissue model to forces. This leads to a system of coupled second order ODEs that relate the node positions, velocities, and accelerations to the nodal forces. The equation for node i is

$$m_i \frac{d^2\mathbf{x}_i}{dt^2} + \gamma_i \frac{d\mathbf{x}_i}{dt} + \tilde{\mathbf{g}}_i + \tilde{\mathbf{q}}_i + \tilde{\mathbf{s}}_i + \tilde{\mathbf{h}}_i = \tilde{\mathbf{f}}_i$$

- m_i is the nodal mass,
- γ_i is the damping coefficient,
- $\tilde{\mathbf{g}}_i$ is the total spring force at node i,
- $\tilde{\mathbf{q}}_i$ is the total volume preservation force at node i,
- $\tilde{\mathbf{s}}_i$ is the total skull penetration force at node i,
- $\tilde{\mathbf{h}}_i$ is the total nodal restoration force at node i,
- $\tilde{\mathbf{f}}_i$ is the total applied muscle force at node i,

3.9 Numerical Simulation

The solution to the above system of ODEs is approximated by using the well-known, explicit Euler method. At each iteration, the nodal acceleration at time t is computed by dividing the net force by nodal mass. The nodal velocity is then calculated by integrating once, and another integration is done to compute the nodal positions at the next time step $t + \Delta t$, as follows:

$$\begin{aligned} \mathbf{a}_i^t &= \frac{1}{m_i}(\tilde{\mathbf{f}}_i^t - \gamma_i \mathbf{v}_i^t - \tilde{\mathbf{g}}_i^t - \tilde{\mathbf{q}}_i^t - \tilde{\mathbf{s}}_i^t - \tilde{\mathbf{h}}_i^t) \\ \mathbf{v}_i^{t+\Delta t} &= \mathbf{v}_i^t + \Delta t \mathbf{a}_i^t \\ \mathbf{x}_i^{t+\Delta t} &= \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t} \end{aligned}$$

3.10 Default Parameters

The default parameters for the physical/numerical simulation and the spring stiffness values of different layers are as follows:

Mass (m)	Time step (Δt)	Damping (γ)
0.5	0.01	30

	Epid	Derm-fat 1	Derm-fat 2	Fascia	Muscle
c	60	30	70	80	10

3.11 Parallel Processing for Facial Animation

The explicit Euler method allows us to easily carry out the numerical simulation of the dynamic skin/muscle model in parallel. This is because at each time step all the calculations are based on the results from the previous time step. Therefore, parallelization is achieved by evenly distributing calculations at each time step to all available processors. This parallel approach increases the animation speed to allow us to simulate facial expressions at interactive rates on our Silicon Graphics multiprocessor workstation.

4 Geometry Models for Other Head Components

To complete our physics-based face model, additional geometric models are combined along with the skin/muscle/skull models developed in the previous section. These include the eyes, eyelids, teeth, neck, hair, and bust (Figure 10). See Plate 5 for an example of a complete model.

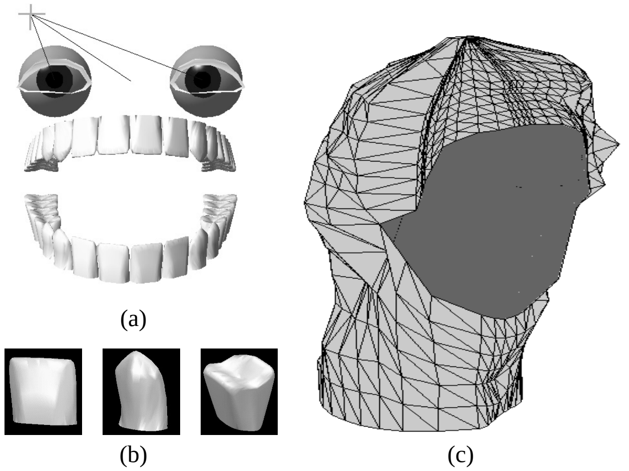

(a)
(b)
(c)

Figure 10: (a) Geometric models of eyes, eyelids, and teeth (b) Incisor, canine, and molar teeth. (c) hair and neck.

4.1 Eyes

Eyes are constructed from spheres with adjustable irises and adjustable pupils (Figure 10(a)). The eyes are automatically scaled to fit the facial model and are positioned into it. The eyes rotate kinematically in a coordinated fashion so that they will always converge on a specified fixation point in three-dimensional space that defines the field of view. Through a simple illumination computation, the eyes can automatically dilate and contract the pupil size in accordance with the amount of light entering the eye.

4.2 Eyelids

The eyelids are polygonal models which can blink kinematically during animation (see Figure 10(a)). Note that the eyelids are open in Figure 10(a).

If the subject is scanned with open eyes, the sensor will not observe the eyelid texture. An eyelid texture is synthesized by a relaxation based interpolation algorithm similar to the one described in section 2.1. The relaxation algorithm interpolates a suitable eyelid texture from the immediately surrounding texture map. Figure 11 shows the results of the eyelid texture interpolation.

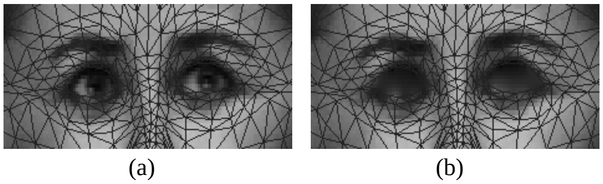

(a)
(b)

Figure 11: (a) Face texture image with adapted mesh before eyelid texture synthesis (b) after eyelid texture synthesis.

4.3 Teeth

We have constructed a full set of generic teeth based on dental images. Each tooth is a NURBS surfaces of degree 2. Three different teeth shapes, the incisor, canine, and molar, are modeled (Figure 10(b)). We use different orientations and scalings of these basic shapes to model the full set of upper and lower teeth shown in Figure 10(a). The dentures are automatically scaled to fit in length, curvature, etc., and are positioned behind the mouth of the facial model.

4.4 Hair, Neck, and Bust Geometry

The hair and bust are both rigid polygonal models (see Figure 10(c)). They are modeled from the range data directly, by extending the facial mesh in a predetermined fashion to the boundaries of the range and reflectance data, and sampling the images as before.

The neck can be twisted, bent and rotated with three degrees of freedom. See Figure 12 for illustrations of the possible neck articulations.

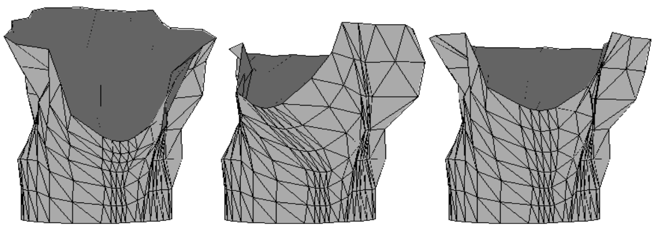


Figure 12: articulation of neck.

5 Animation Examples

Plate 1 illustrates several examples of animating the physics-based face model after conformation to the “Heidi” scanned data (see Plate 2).

- The *surprise* expression results from contraction of the outer frontalis, major frontalis, inner frontalis, zygomatics major, zygomatics minor, depressor labii, and mentalis, and rotation of the jaw.
- The *anger* expression results from contraction of the corrugator, lateral corrugator, levator labii, levator labii nasi, anguli depressor, depressor labii, and mentalis.
- The *quizzical look* results from an asymmetric contraction of the major frontalis, outer frontalis, corrugator, lateral corrugator, levator labii, and buccinator.
- The *sadness* expression results from a contraction of the inner frontalis, corrugator, lateral corrugator, anguli depressor, and depressor labii.

Plate 6 demonstrates the performance of our face model construction algorithm on two male individuals (“Giovanni” and “Mick”). Note that the algorithm is tolerant of some amount of facial hair.

Plate 7 shows a third individual “George.” Note the image at the lower left, which shows two additional expression effects—cheek puffing, and lip puckering—that combine to simulate the vigorous blowing of air through the lips. The cheek puffing was created by applying outwardly directed radial forces to “inflate” the deformable cheeks. The puckered lips were created by applying radial pursing forces and forward protruding forces to simulate the action of the orbicularis oris sphincter muscle which circles the mouth.

Finally, Plate 8 shows several frames from a two-minute animation “*Bureaucrat Too*” (a second-generation version of the 1990 “*Bureaucrat*” which was animated using the generic facial model in [20]). Here “George” tries to read landmark papers on facial modeling and deformable models in the SIGGRAPH ’87 proceedings, only to realize that he doesn’t yet have a brain!

6 Conclusion and Future Work

The human face consists of a biological tissue layer with nonlinear deformation properties, a muscle layer knit together under the skin, and an impenetrable skull structure beneath the muscle layer. We have presented a physics-based model of the face which takes all of these structures into account. Furthermore, we have demonstrated a new technique for automatically constructing face models of this sort and conforming them to individuals by exploiting high-resolution laser scanner data. The conformation process is carried out by a feature matching algorithm based on a reusable generic mesh. The conformation process, efficiently captures facial geometry and photometry, positions and scales facial muscles, and also estimates the skull structure over which the new synthetic facial tissue model can slide. Our facial modeling approach achieves an unprecedented level of realism and fidelity to any specific individual. It also achieves a good compromise between the complete emulation of the complex biomechanical structures and functionality of the human face and real-time simulation performance on state-of-the-art computer graphics and animation hardware.

Although we formulate the synthetic facial skin as a layered tissue model, our work does not yet exploit knowledge of the variable thickness of the layers in different areas of the face. This issue will in all likelihood be addressed in the future by incorporating additional input data about the subject acquired using noninvasive medical scanners such as CT or MR.

Acknowledgments

The authors thank Lisa White and Jim Randall for developing the piecewise linear muscle model used to model the mouth. Range/RGB facial data were provided courtesy of Cyberware, Inc., Monterey, CA. The first two authors thank the Natural Science and Engineering Research Council of Canada for financial support. DT is a fellow of the Canadian Institute for Advanced Research.

References

[1] T. Akimoto, Y. Suenaga, and R. Wallace. Automatic creation of 3D facial models. *IEEE Computer Graphics and Applications*, 13(5):16–22, September 1993.

[2] James Doyle and James Philips. *Manual on Experimental Stress Analysis*. Society for Experimental Mechanics, fifth edition, 1989.

[3] Irfan A. Essa. *Visual Interpretation of Facial Expressions using Dynamic Modeling*. PhD thesis, MIT, 1994.

[4] Frick and Hans. *Human Anatomy*, volume 1. Thieme Medical Publishers, Stuttgart, 1991.

[5] H. Gray. *Anatomy of the Human Body*. Lea & Febiber, Philadelphia, PA, 29th edition, 1985.

[6] Brian Guenter. A system for simulating human facial expression. In *State of the Art in Computer Animation*, pages 191–202. Springer-Verlag, 1992.

[7] T. Kurihara and K. Arai. A transformation method for modeling and animation of the human face from photographs. In *State of the Art in Computer Animation*, pages 45–57. Springer-Verlag, 1991.

[8] Y.C. Lee, D. Terzopoulos, and K. Waters. Constructing physics-based facial models of individuals. In *Proceedings of Graphics Interface '93*, pages 1–8, Toronto, May 1993.

[9] N. Magneneat-Thalmann, H. Minh, M. Angelis, and D. Thalmann. Design, transformation and animation of human faces. *Visual Computer*, 5:32–39, 1989.

[10] D. Metaxas and E. Milios. Reconstruction of a color image from nonuniformly distributed sparse and noisy data. *Computer Vision, Graphics, and Image Processing*, 54(2):103–111, March 1992.

[11] M. Nahas, H. Hutric, M. Rioux, and J. Domey. Facial image synthesis using skin texture recording. *Visual Computer*, 6(6):337–343, 1990.

[12] M. Oka, K. Tsutsui, A. Ohba, Y. Kurauchi, and T. Tago. Real-time manipulation of texture-mapped surfaces. In *SIGGRAPH 21*, pages 181–188. ACM Computer Graphics, 1987.

[13] F. Parke. Computer generated animation of faces. In *ACM National Conference*, pages 451–457. ACM, 1972.

[14] F. Parke. Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, November 1982.

[15] F. Parke. Parameterized models for facial animation revisited. In *SIGGRAPH Facial Animation Tutorial Notes*, pages 43–56. ACM SIGGRAPH, 1989.

[16] Elizabeth C. Patterson, Peter C. Litwinowicz, and N. Greene. Facial animation by spatial mapping. In *State of the Art in Computer Animation*, pages 31–44. Springer-Verlag, 1991.

[17] S. Platt and N. Badler. Animating facial expression. *Computer Graphics*, 15(3):245–252, August 1981.

[18] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438, 1988.

[19] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *Proceedings of Computer Vision and Pattern Recognition Conference*, pages 70–75. IEEE, June 1991.

[20] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis, and animation. *Visualization and Computer Animation*, 1:73–80, 1990.

[21] K. Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics*, 22(4):17–24, 1987.

[22] K. Waters. A physcial model of facial tissue and muscle articulation derived from computer tomography data. In *Visualization in Biomedical Computing*, pages 574–583. SPIE, Vol. 1808, 1992.

[23] K. Waters and D. Terzopoulos. Modeling and animating faces using scanned data. *Visualization and Computer Animation*, 2:123–128, 1991.

[24] L. Williams. Performance-driven facial animation. In *SIGGRAPH 24*, pages 235–242. ACM Computer Graphics, 1990.

[25] J. Yau and N. Duffy. 3-D facial animation using image samples. In *New Trends in Computer Graphics*, pages 64–73. Springer-Verlag, 1988.

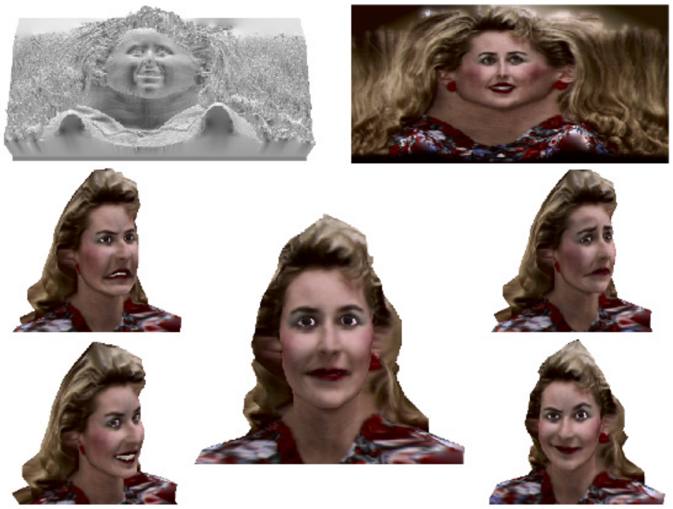


Plate 1: Objective. *Input*: Range map in 3D and texture map (top). *Output*: Functional face model for animation.

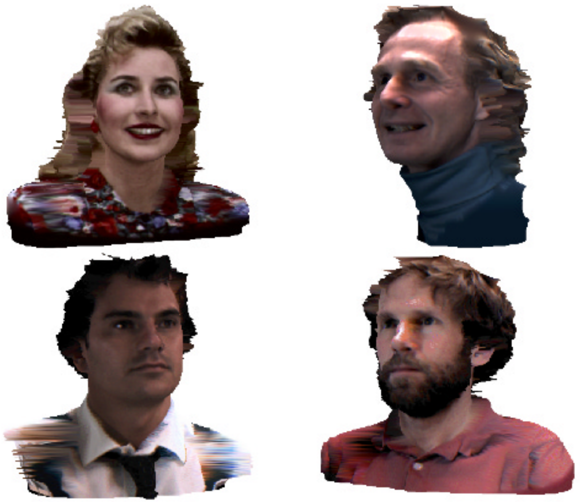


Plate 2: Raw 512 × 256 digitized data for Heidi (top left), George (top right), Giovanni (bottom left), Mick (bottom right).

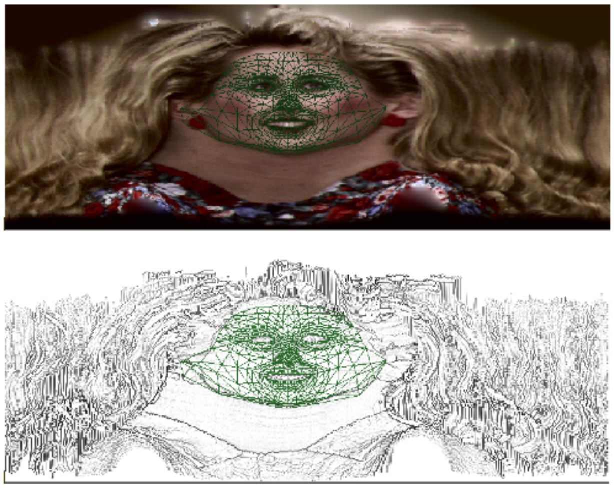


Plate 3: Adapted face mesh overlaying texture map and Laplacian filtered range map of Heidi.

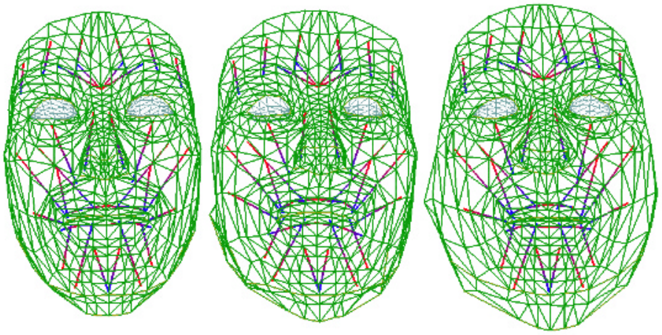


Plate 4: Muscle fiber vector embedded in generic face model and two adapted faces of Heidi and George.

Plate 5: Complete, functional head model of Heidi with physics-based face and geometric eyes, teeth, hair, neck, and shoulders (in Monument Valley).

Plate 6: Animation examples of Giovanni and Mick.

Plate 7: Animation example of George.

COMPUTER GRAPHICS
COMPUTER GRAPHICS

Plate 8: George in four scenes from “*Bureaucrat Too*”.

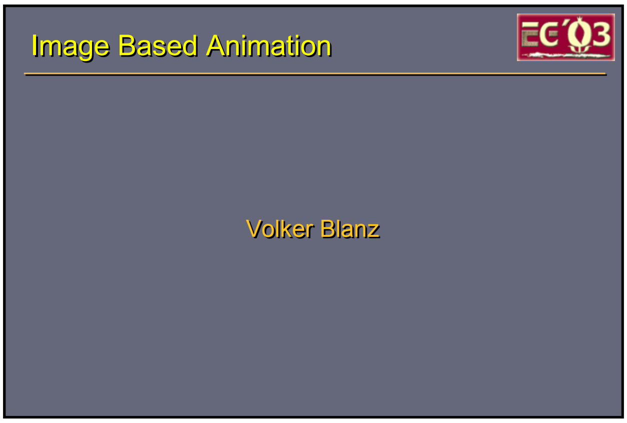
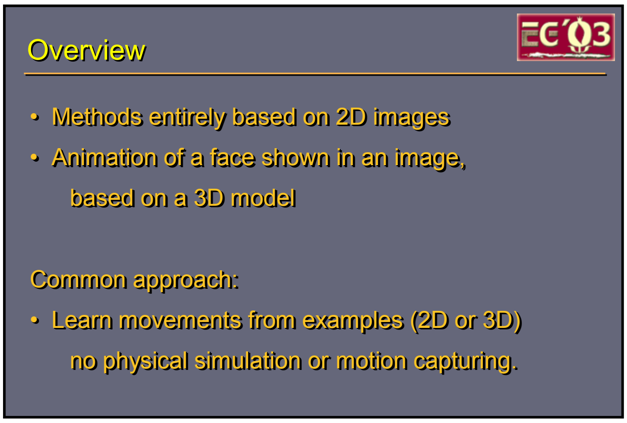


Image Based Animation

Volker Blanz

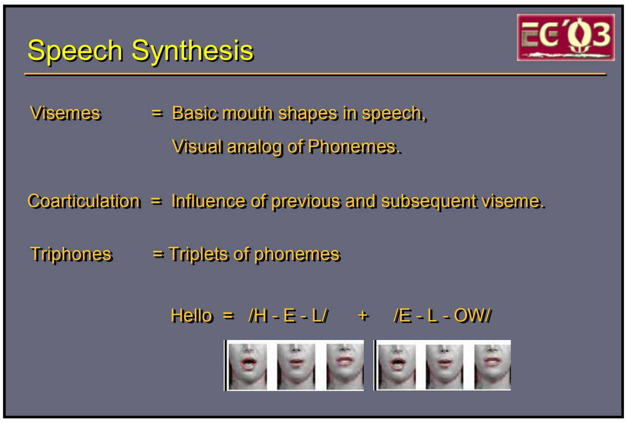


Overview

- Methods entirely based on 2D images
- Animation of a face shown in an image,
  based on a 3D model

Common approach:

- Learn movements from examples (2D or 3D)
  no physical simulation or motion capturing.



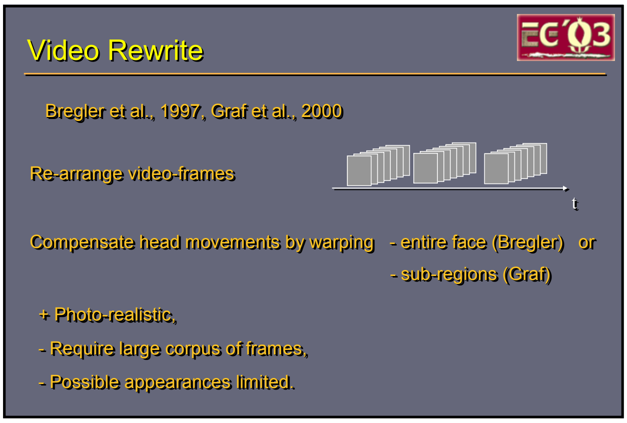
Speech Synthesis

Visemes = Basic mouth shapes in speech,
Visual analog of Phonemes.

Coarticulation = Influence of previous and subsequent viseme.

Triphones = Triplets of phonemes

Hello = /H - E - L/ + /E - L - OW/



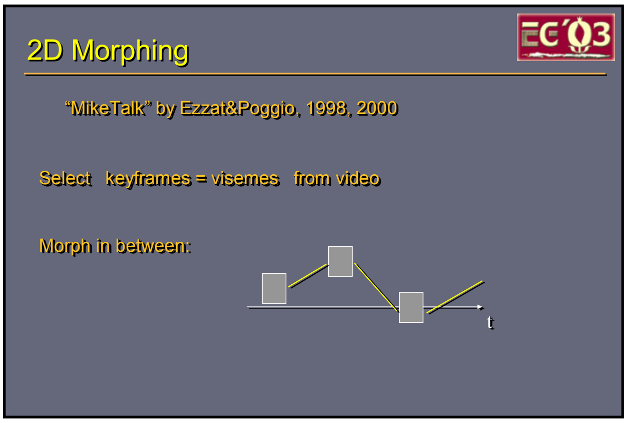
Video Rewrite

Bregler et al., 1997, Graf et al., 2000

Re-arrange video-frames

Compensate head movements by warping - entire face (Bregler) or
- sub-regions (Graf)

+ Photo-realistic,
- Require large corpus of frames,
- Possible appearances limited.

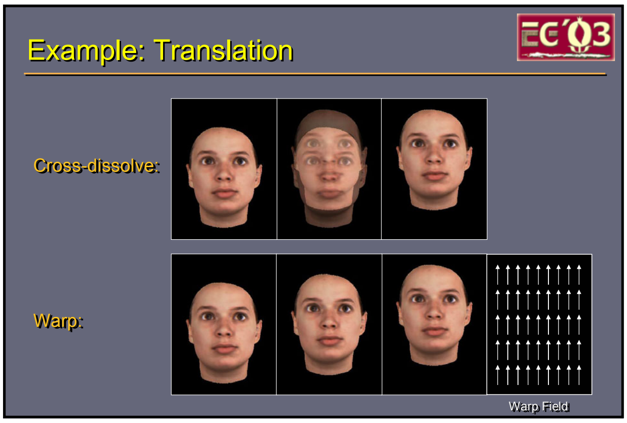


2D Morphing

“MikeTalk” by Ezzat&Poggio, 1998, 2000

Select keyframes = visemes from video

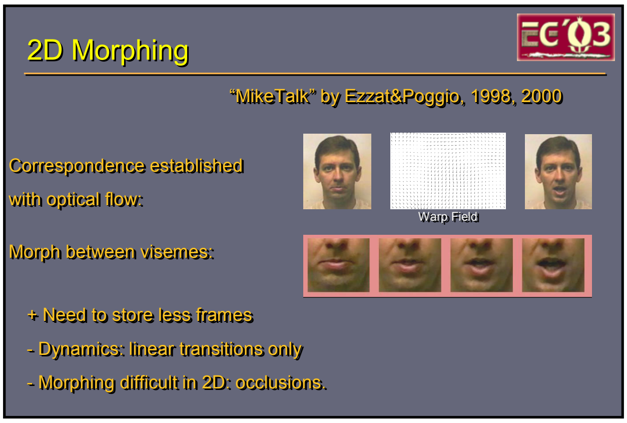
Morph in between:



Example: Translation

Cross-dissolve:

Warp:



2D Morphing

“MikeTalk” by Ezzat&Poggio, 1998, 2000

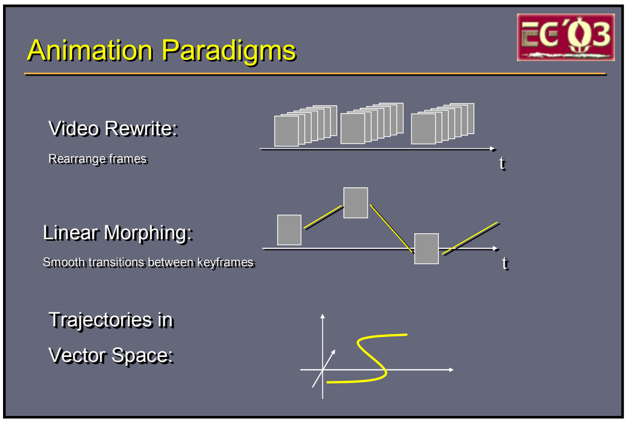
Correspondence established with optical flow:

Morph between visemes:

+ Need to store less frames

- Dynamics: linear transitions only

- Morphing difficult in 2D: occlusions.



Animation Paradigms

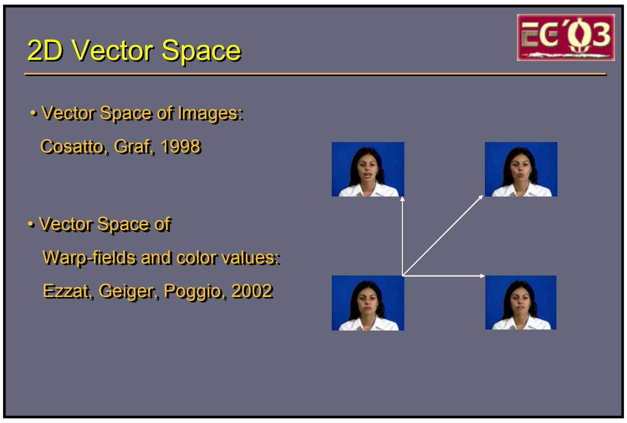
Video Rewrite:

Rearrange frames

Linear Morphing:

Smooth transitions between keyframes

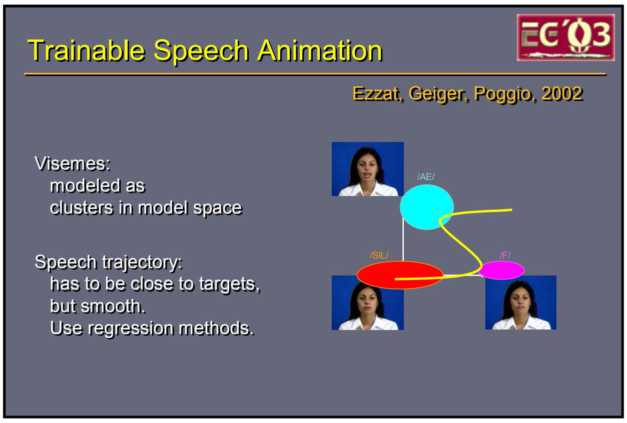
Trajectories in Vector Space:



2D Vector Space

- Vector Space of Images:
  Cosatto, Graf, 1998

- Vector Space of
  Warp-fields and color values:
  Ezzat, Geiger, Poggio, 2002

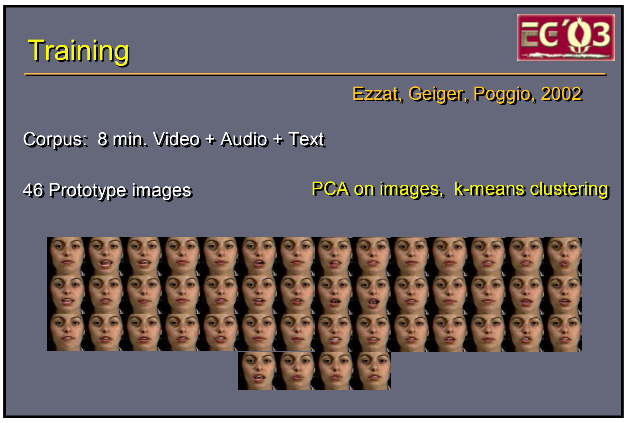


Trainable Speech Animation

Ezzat, Geiger, Poggio, 2002

Visemes:
modeled as
clusters in model space

Speech trajectory:
has to be close to targets,
but smooth.
Use regression methods.

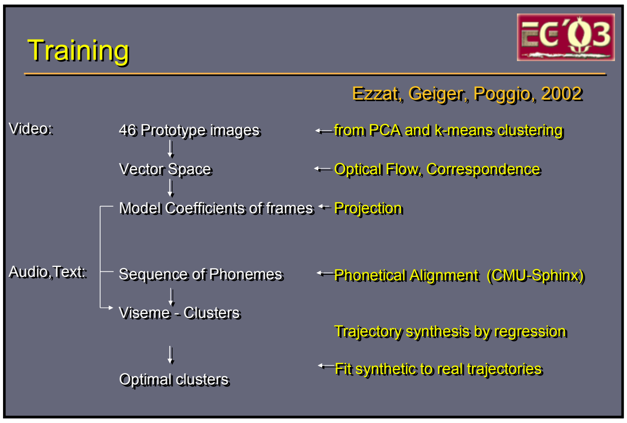


Training

Ezzat, Geiger, Poggio, 2002

Corpus: 8 min. Video + Audio + Text

46 Prototype images PCA on images, k-means clustering



Training

Ezzat, Geiger, Poggio, 2002

Video: 46 Prototype images ← from PCA and k-means clustering

Vector Space ← Optical Flow, Correspondence

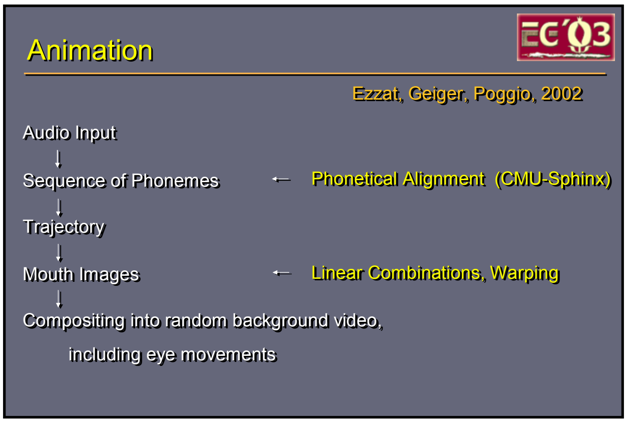
Model Coefficients of frames ← Projection

Audio,Text: Sequence of Phonemes ← Phonetical Alignment (CMU-Sphinx)

Viseme - Clusters

Trajectory synthesis by regression

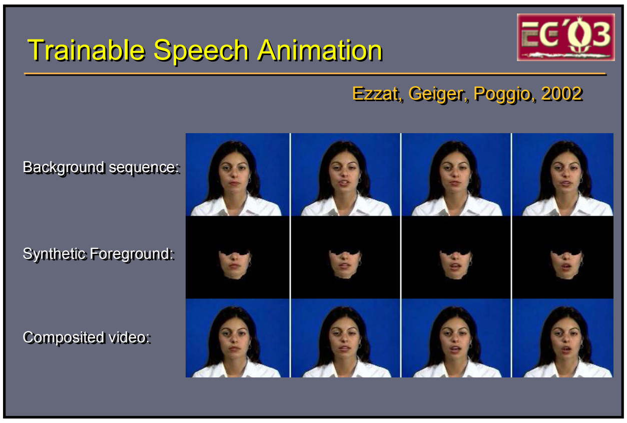
Optimal clusters ← Fit synthetic to real trajectories



Animation

Ezzat, Geiger, Poggio, 2002

Audio Input
↓
Sequence of Phonemes ← Phonetical Alignment (CMU-Sphinx)
↓
Trajectory
↓
Mouth Images ← Linear Combinations, Warping
↓
Compositing into random background video,
including eye movements



Trainable Speech Animation

Ezzat, Geiger, Poggio, 2002

Background sequence:

Synthetic Foreground:

Composited video:

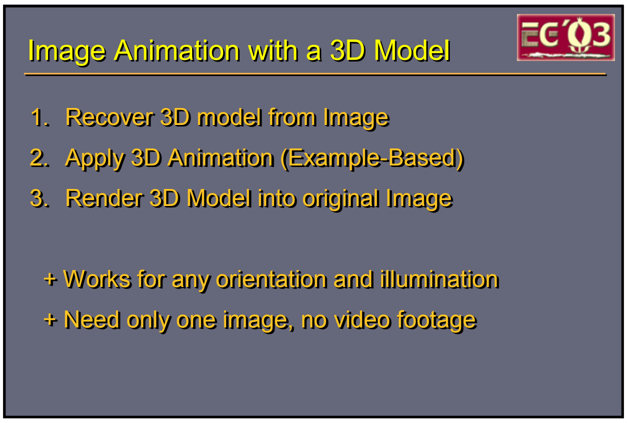
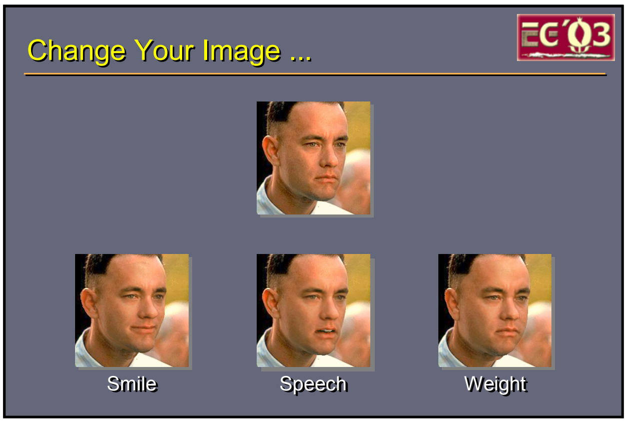


Image Animation with a 3D Model

1. Recover 3D model from Image
2. Apply 3D Animation (Example-Based)
3. Render 3D Model into original Image

+ Works for any orientation and illumination

+ Need only one image, no video footage

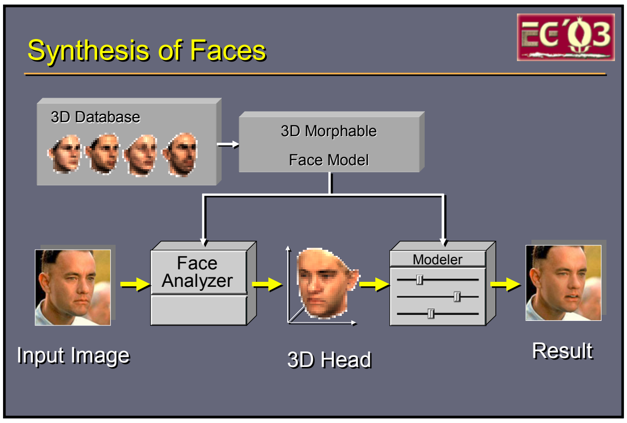


Change Your Image ...

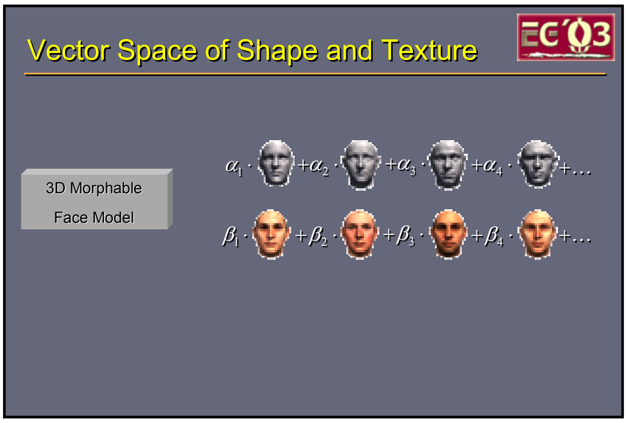
Smile

Speech

Weight



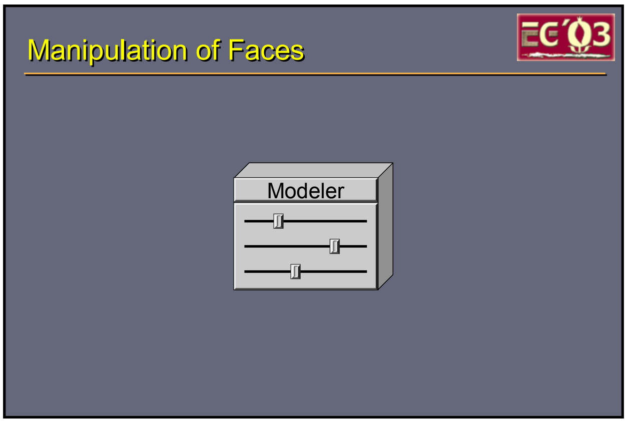
Synthesis of Faces



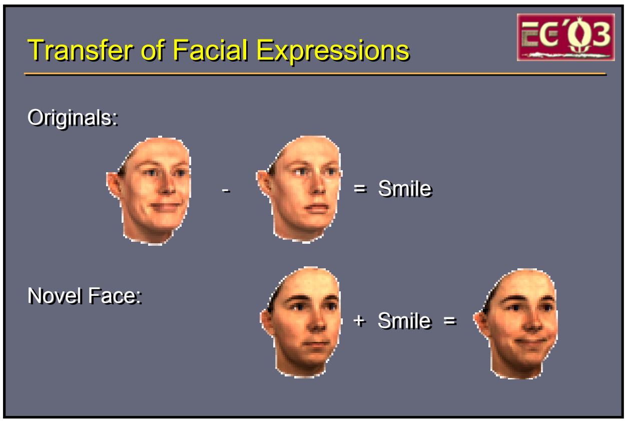
Vector Space of Shape and Texture

$\alpha_1 \cdot \; + \alpha_2 \cdot \; + \alpha_3 \cdot \; + \alpha_4 \cdot \; + \ldots$

$\beta_1 \cdot \; + \beta_2 \cdot \; + \beta_3 \cdot \; + \beta_4 \cdot \; + \ldots$



Manipulation of Faces



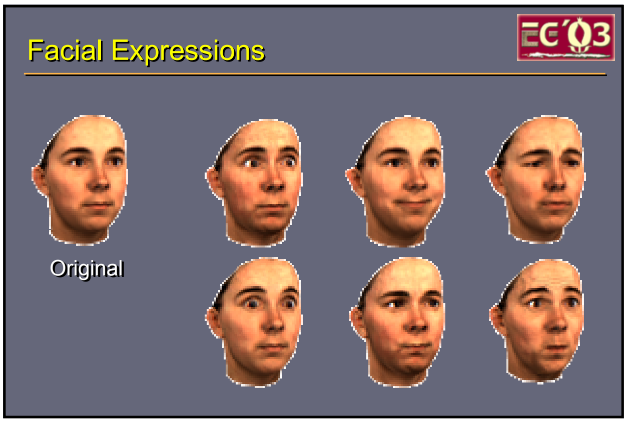
Transfer of Facial Expressions

Originals:

- = Smile

Novel Face:

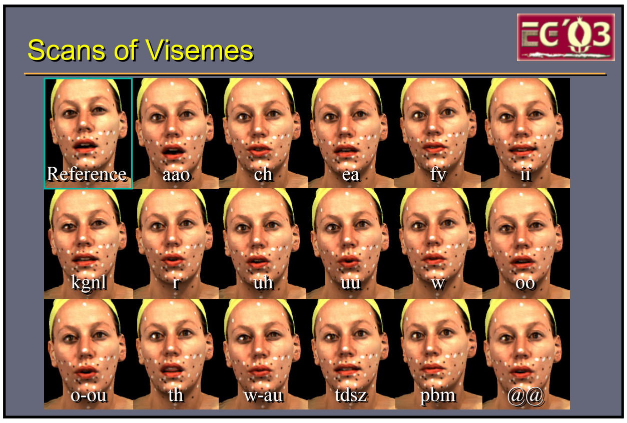
+ Smile =



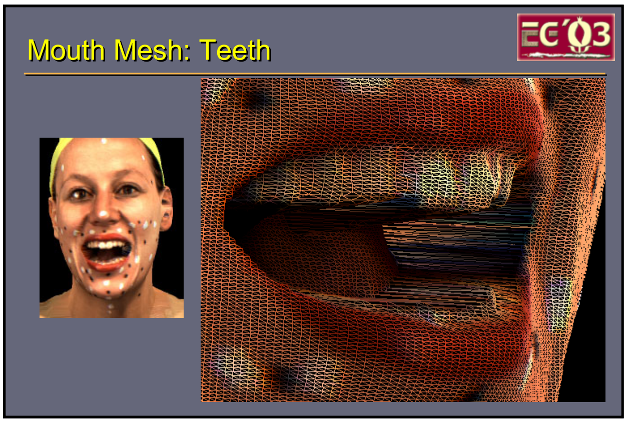
Facial Expressions

Open Mouth

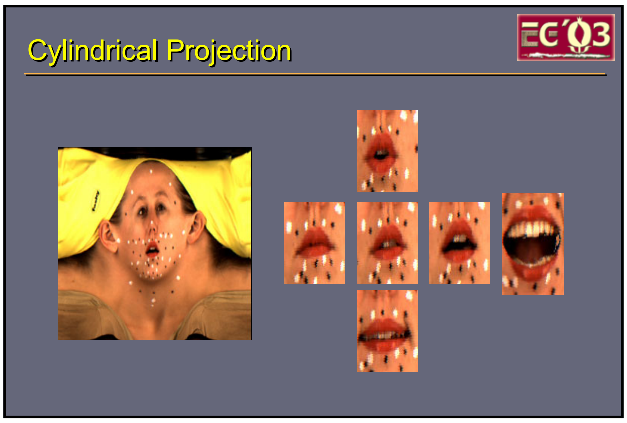
- Mouth poses are learned from static scans.
- New reference scan with open mouth and teeth. Closing the mouth will occlude teeth in 3D.
- Upper jaw teeth remain fixed relative to the head.
- Lower jaw teeth move with tip of chin.



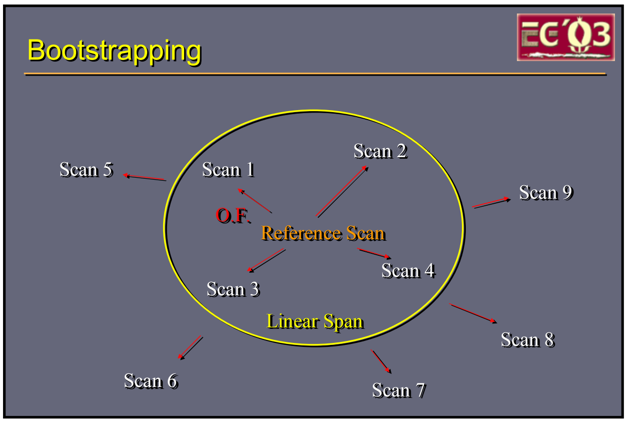
Scans of Visemes



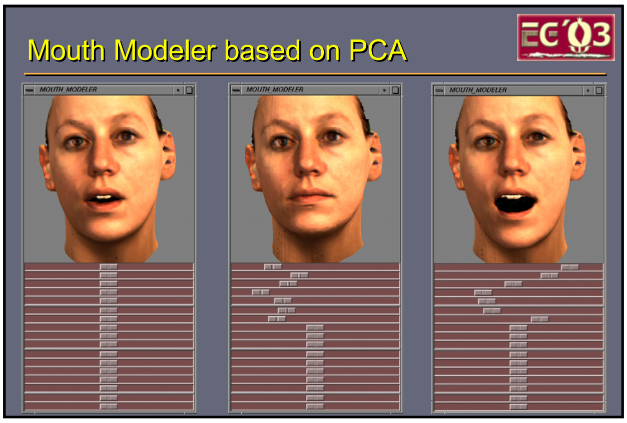
Mouth Mesh: Teeth



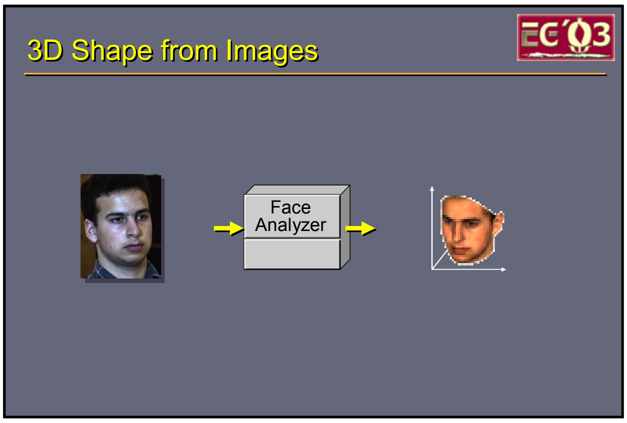
Cylindrical Projection



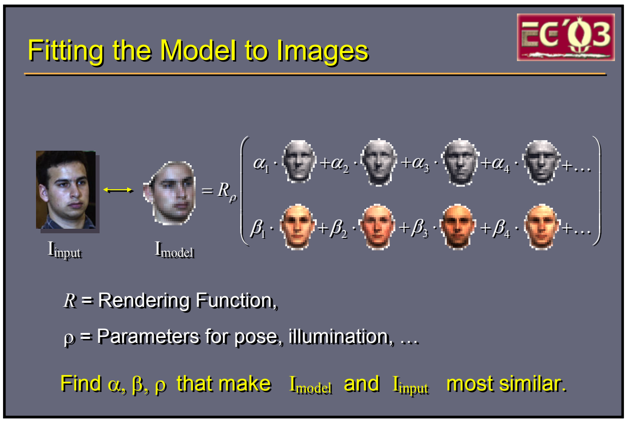
Bootstrapping



Mouth Modeler based on PCA



3D Shape from Images



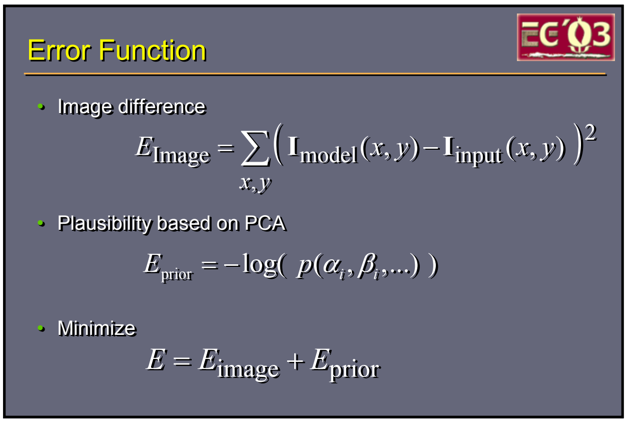
Fitting the Model to Images

$\mathrm{I}_{\mathrm{input}}$
$\mathrm{I}_{\mathrm{model}}$
$= R_\rho$
$\alpha_1 \cdot$ $+\alpha_2 \cdot$ $+\alpha_3 \cdot$ $+\alpha_4 \cdot$ $+\ldots$
$\beta_1 \cdot$ $+\beta_2 \cdot$ $+\beta_3 \cdot$ $+\beta_4 \cdot$ $+\ldots$

R = Rendering Function,

ρ = Parameters for pose, illumination, …

Find α, β, ρ that make $\mathrm{I}_{\mathrm{model}}$ and $\mathrm{I}_{\mathrm{input}}$ most similar.



Error Function

- Image difference

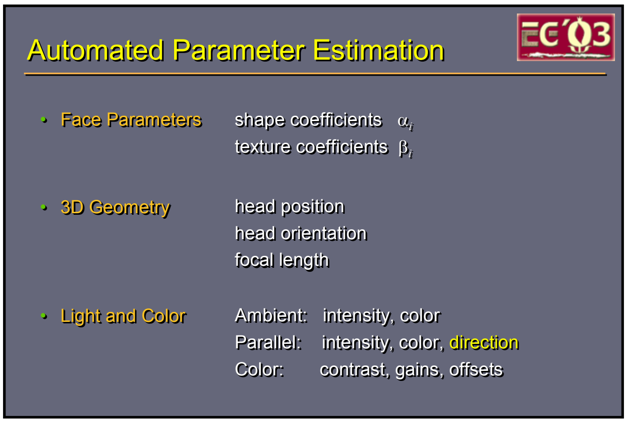
$$E_{\mathrm{Image}} = \sum_{x,y} \left(\mathbf{I}_{\mathrm{model}}(x,y) - \mathbf{I}_{\mathrm{input}}(x,y) \right)^2$$

- Plausibility based on PCA

$$E_{\mathrm{prior}} = -\log(\ p(\alpha_i, \beta_i, \ldots)\ )$$

- Minimize

$$E = E_{\mathrm{image}} + E_{\mathrm{prior}}$$



Automated Parameter Estimation

- Face Parameters — shape coefficients α_i; texture coefficients β_i
- 3D Geometry — head position; head orientation; focal length
- Light and Color — Ambient: intensity, color; Parallel: intensity, color, direction; Color: contrast, gains, offsets

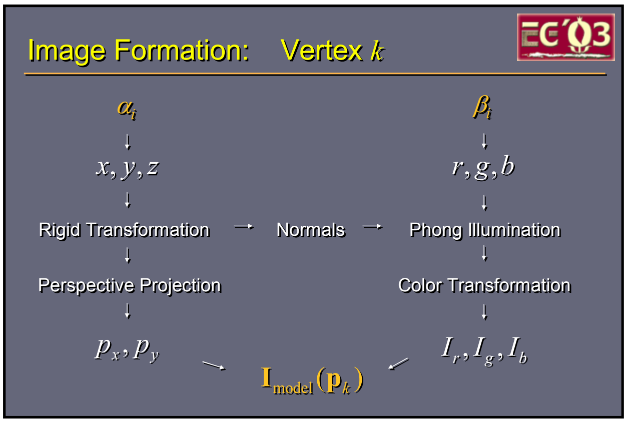


Image Formation: Vertex k

$$\begin{tikzcd}
\alpha_i \arrow[d] & & \beta_i \arrow[d] \\
x,y,z \arrow[d] & & r,g,b \arrow[d] \\
\text{Rigid Transformation} \arrow[r] \arrow[d] & \text{Normals} \arrow[r] & \text{Phong Illumination} \arrow[d] \\
\text{Perspective Projection} \arrow[d] & & \text{Color Transformation} \arrow[d] \\
p_x, p_y \arrow[rd] & & I_r, I_g, I_b \arrow[ld] \\
& \mathbf{I}_{\text{model}}(\mathbf{p}_k) &
\end{tikzcd}$$

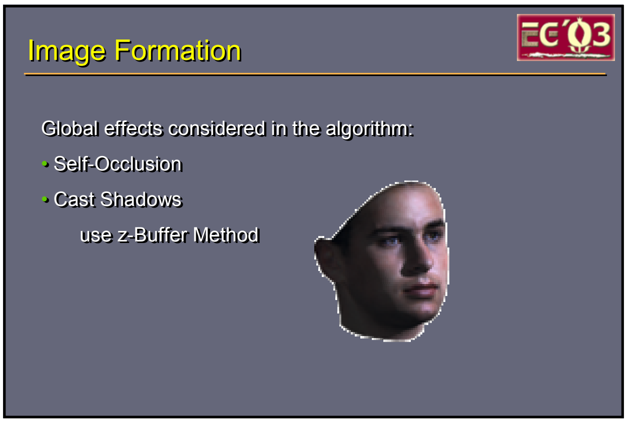
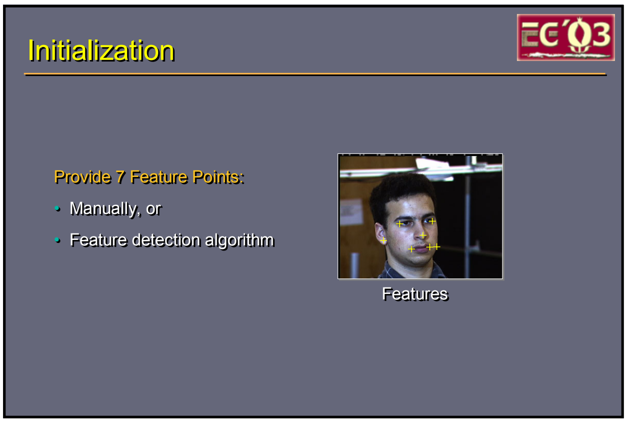


Image Formation

Global effects considered in the algorithm:

- Self-Occlusion
- Cast Shadows

use z-Buffer Method

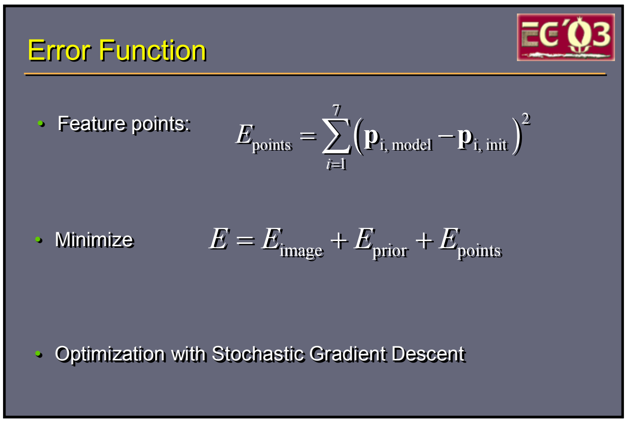


Initialization

Provide 7 Feature Points:

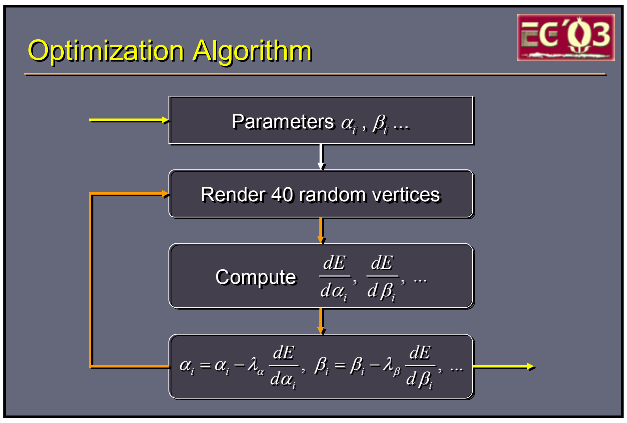
- Manually, or
- Feature detection algorithm

Features



Error Function

- Feature points: $$E_{\text{points}} = \sum_{i=1}^{7} \left(\mathbf{p}_{\text{i, model}} - \mathbf{p}_{\text{i, init}}\right)^2$$
- Minimize $$E = E_{\text{image}} + E_{\text{prior}} + E_{\text{points}}$$
- Optimization with Stochastic Gradient Descent



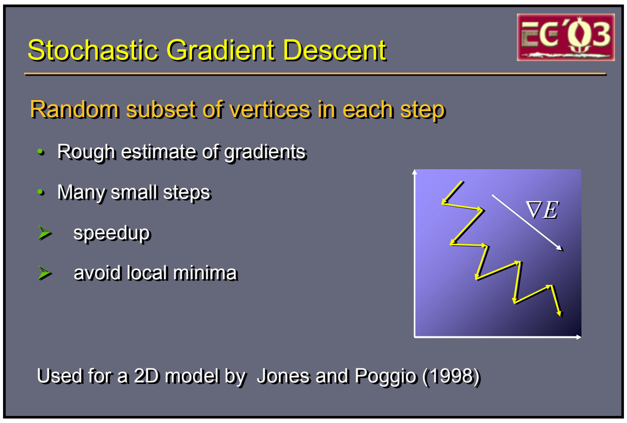
Optimization Algorithm

Parameters α_i , β_i ...

Render 40 random vertices

Compute $\frac{dE}{d\alpha_i}, \frac{dE}{d\beta_i}, \ldots$

$\alpha_i = \alpha_i - \lambda_\alpha \frac{dE}{d\alpha_i}, \; \beta_i = \beta_i - \lambda_\beta \frac{dE}{d\beta_i}, \; \ldots$

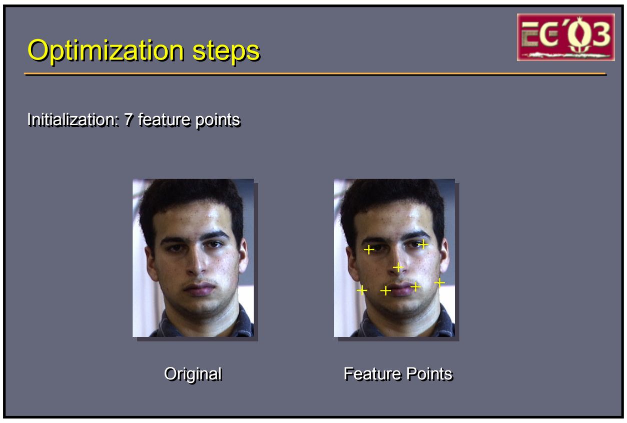


Stochastic Gradient Descent

Random subset of vertices in each step

- Rough estimate of gradients
- Many small steps
  - speedup
  - avoid local minima

Used for a 2D model by Jones and Poggio (1998)

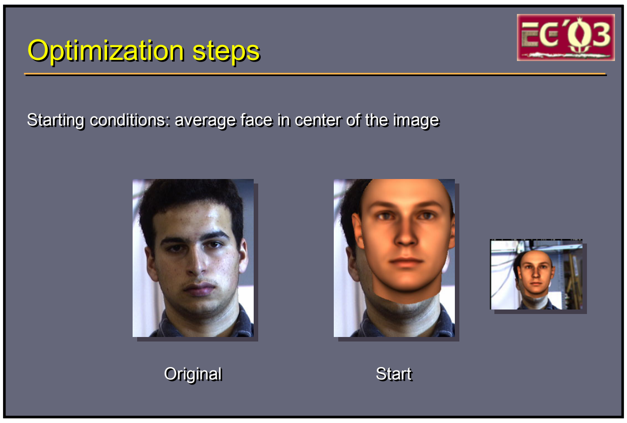


Optimization steps

Initialization: 7 feature points

Original

Feature Points

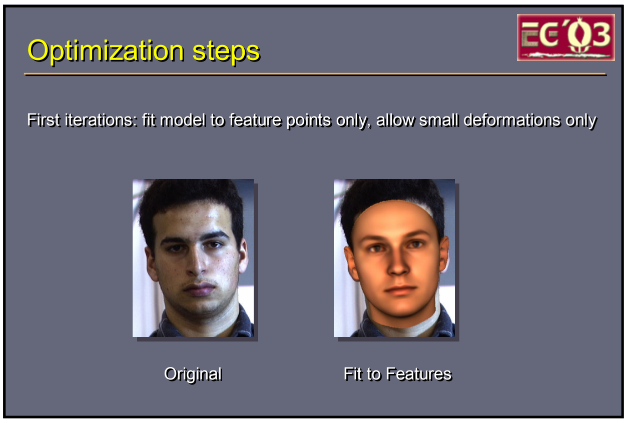


Optimization steps

Starting conditions: average face in center of the image

Original

Start

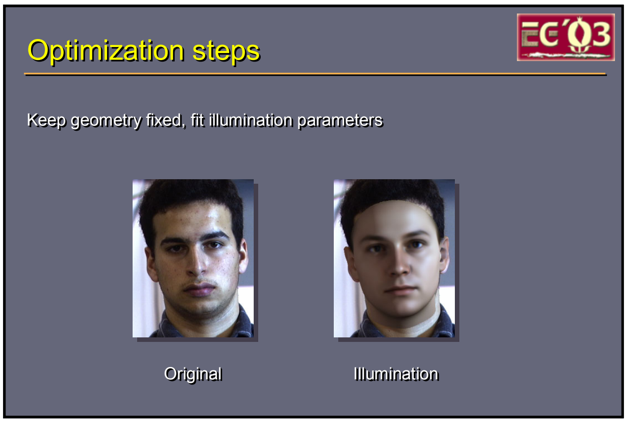


Optimization steps

First iterations: fit model to feature points only, allow small deformations only

Original

Fit to Features

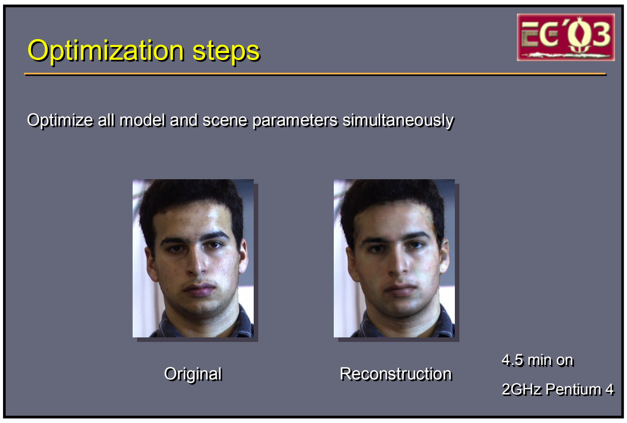


Optimization steps

Keep geometry fixed, fit illumination parameters

Original

Illumination



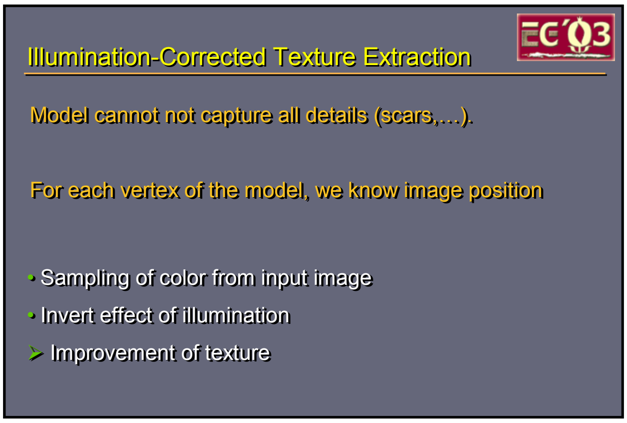
Optimization steps

Optimize all model and scene parameters simultaneously

Original

Reconstruction

4.5 min on
2GHz Pentium 4

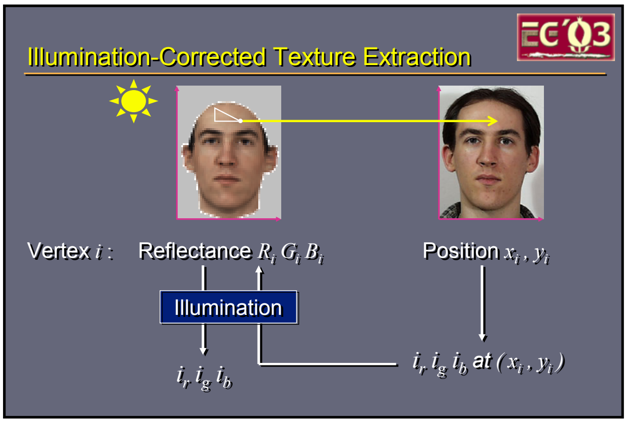


Illumination-Corrected Texture Extraction

Model cannot not capture all details (scars,…).

For each vertex of the model, we know image position

- Sampling of color from input image
- Invert effect of illumination
- ➢ Improvement of texture



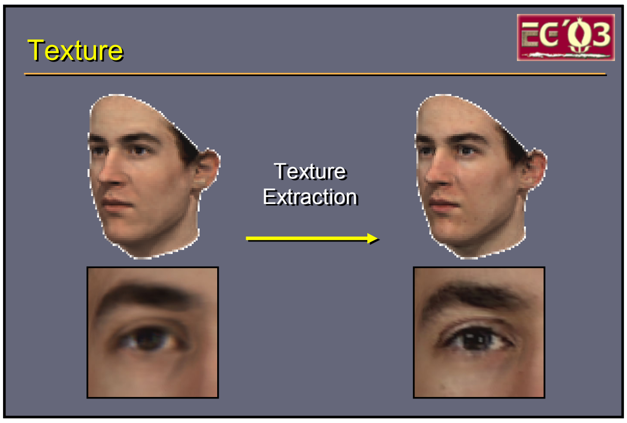
Illumination-Corrected Texture Extraction

Vertex i : Reflectance $R_i\ G_i\ B_i$ Position x_i , y_i

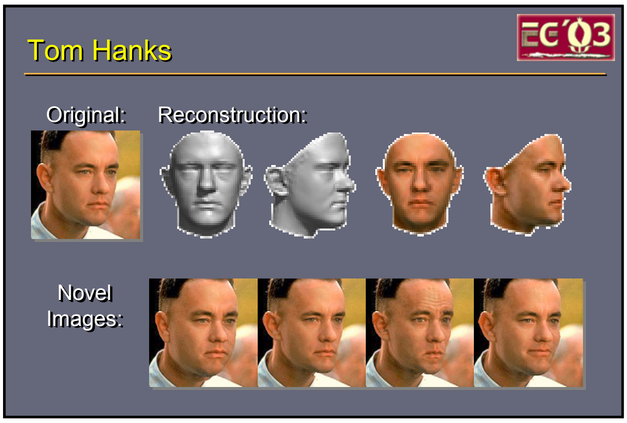
Illumination

$i_r\ i_g\ i_b$

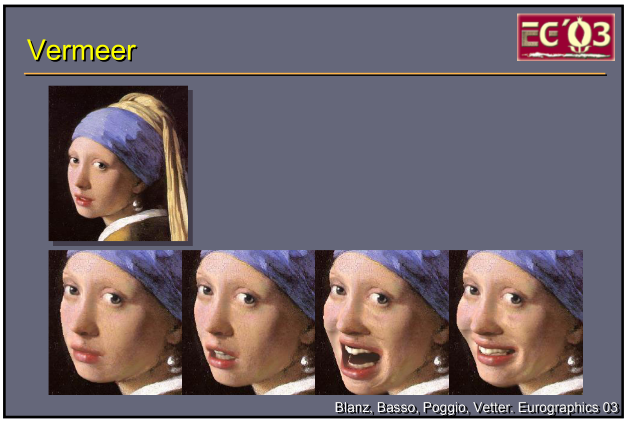
$i_r\ i_g\ i_b$ at (x_i , y_i)



Texture

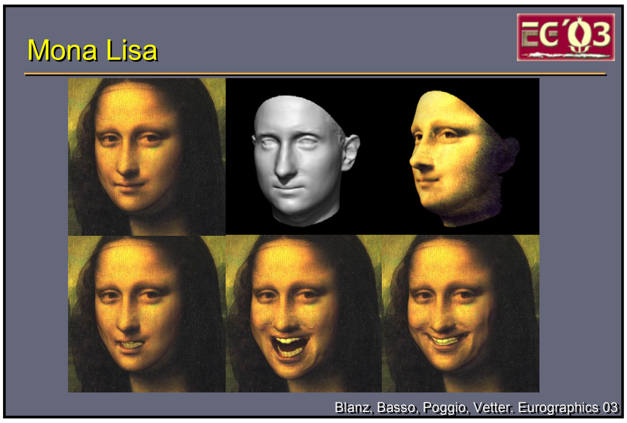


Tom Hanks



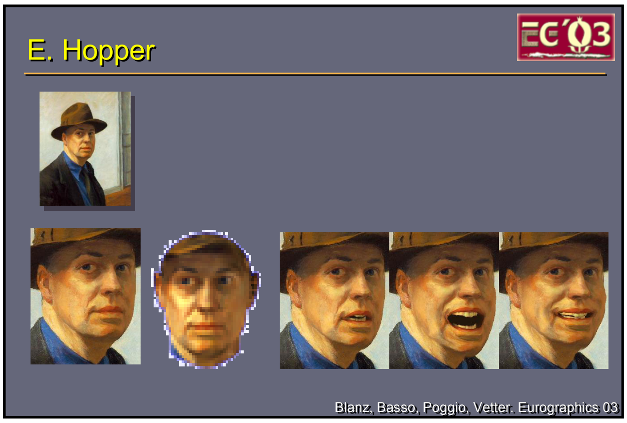
Vermeer

Blanz, Basso, Poggio, Vetter. Eurographics 03



Mona Lisa

Blanz, Basso, Poggio, Vetter. Eurographics 03



E. Hopper

Blanz, Basso, Poggio, Vetter. Eurographics 03

A Morphable Model For The Synthesis Of 3D Faces

Volker Blanz Thomas Vetter

Max-Planck-Institut für biologische Kybernetik,
Tübingen, Germany*

Abstract

In this paper, a new technique for modeling textured 3D faces is introduced. 3D faces can either be generated automatically from one or more photographs, or modeled directly through an intuitive user interface. Users are assisted in two key problems of computer aided face modeling. First, new face images or new 3D face models can be registered automatically by computing dense one-to-one correspondence to an internal face model. Second, the approach regulates the naturalness of modeled faces avoiding faces with an "unlikely" appearance.

Starting from an example set of 3D face models, we derive a morphable face model by transforming the shape and texture of the examples into a vector space representation. New faces and expressions can be modeled by forming linear combinations of the prototypes. Shape and texture constraints derived from the statistics of our example faces are used to guide manual modeling or automated matching algorithms.

We show 3D face reconstructions from single images and their applications for photo-realistic image manipulations. We also demonstrate face manipulations according to complex parameters such as gender, fullness of a face or its distinctiveness.

Keywords: facial modeling, registration, photogrammetry, morphing, facial animation, computer vision

1 Introduction

Computer aided modeling of human faces still requires a great deal of expertise and manual control to avoid unrealistic, non-face-like results. Most limitations of automated techniques for face synthesis, face animation or for general changes in the appearance of an individual face can be described either as the problem of finding corresponding feature locations in different faces or as the problem of separating realistic faces from faces that could never appear in the real world. The correspondence problem is crucial for all morphing techniques, both for the application of motion-capture data to pictures or 3D face models, and for most 3D face reconstruction techniques from images. A limited number of labeled feature points marked in one face, e.g., the tip of the nose, the eye corner and less prominent points on the cheek, must be located precisely in another face. The number of manually labeled feature points varies from application to application, but usually ranges from 50 to 300.

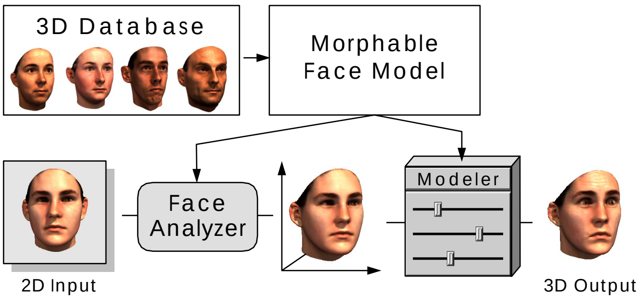


Figure 1: Derived from a dataset of prototypical 3D scans of faces, the morphable face model contributes to two main steps in face manipulation: (1) deriving a 3D face model from a novel image, and (2) modifying shape and texture in a natural way.

Only a correct alignment of all these points allows acceptable intermediate morphs, a convincing mapping of motion data from the reference to a new model, or the adaptation of a 3D face model to 2D images for 'video cloning'. Human knowledge and experience is necessary to compensate for the variations between individual faces and to guarantee a valid location assignment in the different faces. At present, automated matching techniques can be utilized only for very prominent feature points such as the corners of eyes and mouth.

A second type of problem in face modeling is the separation of natural faces from non faces. For this, human knowledge is even more critical. Many applications involve the design of completely new natural looking faces that can occur in the real world but which have no "real" counterpart. Others require the manipulation of an existing face according to changes in age, body weight or simply to emphasize the characteristics of the face. Such tasks usually require time-consuming manual work combined with the skills of an artist.

In this paper, we present a parametric face modeling technique that assists in both problems. First, arbitrary human faces can be created simultaneously controlling the likelihood of the generated faces. Second, the system is able to compute correspondence between new faces. Exploiting the statistics of a large dataset of 3D face scans (geometric and textural data, $Cyberware^{TM}$) we built a morphable face model and recover domain knowledge about face variations by applying pattern classification methods. The morphable face model is a multidimensional 3D morphing function that is based on the linear combination of a large number of 3D face scans. Computing the average face and the main modes of variation in our dataset, a probability distribution is imposed on the morphing function to avoid unlikely faces. We also derive parametric descriptions of face attributes such as gender, distinctiveness, "hooked" noses or the weight of a person, by evaluating the distribution of exemplar faces for each attribute within our face space.

Having constructed a parametric face model that is able to generate almost any face, the correspondence problem turns into a mathematical optimization problem. New faces, images or 3D face scans, can be registered by minimizing the difference between the new face and its reconstruction by the face model function. We devel-

*MPI für biol. Kybernetik, Spemannstr. 38, 72076 Tübingen, Germany. E-mail: {volker.blanz, thomas.vetter}@tuebingen.mpg.de

oped an algorithm that adjusts the model parameters automatically for an optimal reconstruction of the target, requiring only a minimum of manual initialization. The output of the matching procedure is a high quality 3D face model that is in full correspondence with our morphable face model. Consequently all face manipulations parameterized in our model function can be mapped to the target face. The prior knowledge about the shape and texture of faces in general that is captured in our model function is sufficient to make reasonable estimates of the full 3D shape and texture of a face even when only a single picture is available. When applying the method to several images of a person, the reconstructions reach almost the quality of laser scans.

1.1 Previous and related work

Modeling human faces has challenged researchers in computer graphics since its beginning. Since the pioneering work of Parke [25, 26], various techniques have been reported for modeling the geometry of faces [10, 11, 22, 34, 21] and for animating them [28, 14, 19, 32, 22, 38, 29]. A detailed overview can be found in the book of Parke and Waters [24].

The key part of our approach is a generalized model of human faces. Similar to the approach of DeCarlos et al. [10], we restrict the range of allowable faces according to constraints derived from prototypical human faces. However, instead of using a limited set of measurements and proportions between a set of facial landmarks, we directly use the densely sampled geometry of the exemplar faces obtained by laser scanning ($Cyberware^{TM}$). The dense modeling of facial geometry (several thousand vertices per face) leads directly to a triangulation of the surface. Consequently, there is no need for variational surface interpolation techniques [10, 23, 33]. We also added a model of texture variations between faces. The morphable 3D face model is a consequent extension of the interpolation technique between face geometries, as introduced by Parke [26]. Computing correspondence between individual 3D face data automatically, we are able to increase the number of vertices used in the face representation from a few hundreds to tens of thousands. Moreover, we are able to use a higher number of faces, and thus to interpolate between hundreds of 'basis' faces rather than just a few. The goal of such an extended morphable face model is to represent any face as a linear combination of a limited basis set of face prototypes. Representing the face of an arbitrary person as a linear combination (morph) of “prototype” faces was first formulated for image compression in telecommunications [8]. Image-based linear 2D face models that exploit large data sets of prototype faces were developed for face recognition and image coding [4, 18, 37].

Different approaches have been taken to automate the matching step necessary for building up morphable models. One class of techniques is based on optic flow algorithms [5, 4] and another on an active model matching strategy [12, 16]. Combinations of both techniques have been applied to the problem of image matching [36]. In this paper we extend this approach to the problem of matching 3D faces.

The correspondence problem between different three-dimensional face data has been addressed previously by Lee et al.[20]. Their shape-matching algorithm differs significantly from our approach in several respects. First, we compute the correspondence in high resolution, considering shape and texture data simultaneously. Second, instead of using a physical tissue model to constrain the range of allowed mesh deformations, we use the statistics of our example faces to keep deformations plausible. Third, we do not rely on routines that are specifically designed to detect the features exclusively found in faces, e.g., eyes, nose.

Our general matching strategy can be used not only to adapt the morphable model to a 3D face scan, but also to 2D images of faces. Unlike a previous approach [35], the morphable 3D face model is now directly matched to images, avoiding the detour of generating intermediate 2D morphable image models. As a consequence, head orientation, illumination conditions and other parameters can be free variables subject to optimization. It is sufficient to use rough estimates of their values as a starting point of the automated matching procedure.

Most techniques for ‘face cloning’, the reconstruction of a 3D face model from one or more images, still rely on manual assistance for matching a deformable 3D face model to the images [26, 1, 30]. The approach of Pighin et al. [28] demonstrates the high realism that can be achieved for the synthesis of faces and facial expressions from photographs where several images of a face are matched to a single 3D face model. Our automated matching procedure could be used to replace the manual initialization step, where several corresponding features have to be labeled in the presented images.

For the animation of faces, a variety of methods have been proposed. For a complete overview we again refer to the book of Parke and Waters [24]. The techniques can be roughly separated in those that rely on physical modeling of facial muscles [38, 17], and in those applying previously captured facial expressions to a face [25, 3]. These performance based animation techniques compute the correspondence between the different facial expressions of a person by tracking markers glued to the face from image to image. To obtain photo-realistic face animations, up to 182 markers are used [14]. Working directly on faces without markers, our automated approach extends this number to its limit. It matches the full number of vertices available in the face model to images. The resulting dense correspondence fields can even capture changes in wrinkles and map these from one face to another.

1.2 Organization of the paper

We start with a description of the database of 3D face scans from which our morphable model is built.

In Section 3, we introduce the concept of the morphable face model, assuming a set of 3D face scans that are in full correspondence. Exploiting the statistics of a dataset, we derive a parametric description of faces, as well as the range of plausible faces. Additionally, we define facial attributes, such as gender or fullness of faces, in the parameter space of the model.

In Section 4, we describe an algorithm for matching our flexible model to novel images or 3D scans of faces. Along with a 3D reconstruction, the algorithm can compute correspondence, based on the morphable model.

In Section 5, we introduce an iterative method for building a morphable model automatically from a raw data set of 3D face scans when no correspondences between the exemplar faces are available.

2 Database

Laser scans ($Cyberware^{TM}$) of 200 heads of young adults (100 male and 100 female) were used. The laser scans provide head structure data in a cylindrical representation, with radii $r(h, \phi)$ of surface points sampled at 512 equally-spaced angles ϕ, and at 512 equally spaced vertical steps h. Additionally, the RGB-color values $R(h, \phi)$, $G(h, \phi)$,and $B(h, \phi)$, were recorded in the same spatial resolution and were stored in a texture map with 8 bit per channel.

All faces were without makeup, accessories, and facial hair. The subjects were scanned wearing bathing caps, that were removed digitally. Additional automatic pre-processing of the scans, which for most heads required no human interaction, consisted of a vertical cut behind the ears, a horizontal cut to remove the shoulders, and a normalization routine that brought each face to a standard orientation and position in space. The resultant faces were represented by approximately 70,000 vertices and the same number of color values.

3 Morphable 3D Face Model

The morphable model is based on a data set of 3D faces. Morphing between faces requires full correspondence between all of the faces. In this section, we will assume that all exemplar faces are in full correspondence. The algorithm for computing correspondence will be described in Section 5.

We represent the geometry of a face with a shape-vector $S = (X_1, Y_1, Z_1, X_2, ..., Y_n, Z_n)^T \in \Re^{3n}$, that contains the X, Y, Z-coordinates of its n vertices. For simplicity, we assume that the number of valid texture values in the texture map is equal to the number of vertices. We therefore represent the texture of a face by a texture-vector $T = (R_1, G_1, B_1, R_2, ..., G_n, B_n)^T \in \Re^{3n}$, that contains the R, G, B color values of the n corresponding vertices. A morphable face model was then constructed using a data set of m exemplar faces, each represented by its shape-vector S_i and texture-vector T_i. Since we assume all faces in full correspondence (see Section 5), new shapes S_{model} and new textures T_{model} can be expressed in barycentric coordinates as a linear combination of the shapes and textures of the m exemplar faces:

$$\mathbf{S}_{mod} = \sum_{i=1}^{m} a_i \mathbf{S}_i\,, \quad \mathbf{T}_{mod} = \sum_{i=1}^{m} b_i \mathbf{T}_i\,, \quad \sum_{i=1}^{m} a_i = \sum_{i=1}^{m} b_i = 1.$$

We define the morphable model as the set of faces $(S_{mod}(\vec{a}), T_{mod}(\vec{b}))$, parameterized by the coefficients $\vec{a} = (a_1, a_2 ... a_m)^T$ and $\vec{b} = (b_1, b_2 ... b_m)^T$. [1] Arbitrary new faces can be generated by varying the parameters $\vec{a}$ and $\vec{b}$ that control shape and texture.

For a useful face synthesis system, it is important to be able to quantify the results in terms of their plausibility of being faces. We therefore estimated the probability distribution for the coefficients a_i and b_i from our example set of faces. This distribution enables us to control the likelihood of the coefficients a_i and b_i and consequently regulates the likelihood of the appearance of the generated faces.

We fit a multivariate normal distribution to our data set of 200 faces, based on the averages of shape $\overline{S}$ and texture $\overline{T}$ and the covariance matrices C_S and C_T computed over the shape and texture differences $\Delta S_i = S_i - \overline{S}$ and $\Delta T_i = T_i - \overline{T}$.

A common technique for data compression known as Principal Component Analysis (PCA) [15, 31] performs a basis transformation to an orthogonal coordinate system formed by the eigenvectors s_i and t_i of the covariance matrices (in descending order according to their eigenvalues)[2]:

$$S_{model} = \overline{S} + \sum_{i=1}^{m-1} \alpha_i s_i\,, \quad T_{model} = \overline{T} + \sum_{i=1}^{m-1} \beta_i t_i\,, \qquad (1)$$

$\vec{\alpha}, \vec{\beta} \in \Re^{m-1}$. The probability for coefficients $\vec{\alpha}$ is given by

$$p(\vec{\alpha}) \sim exp[-\frac{1}{2} \sum_{i=1}^{m-1} (\alpha_i / \sigma_i)^2], \qquad (2)$$

with σ_i^2 being the eigenvalues of the shape covariance matrix C_S. The probability $p(\vec{\beta})$ is computed similarly.

Segmented morphable model: The morphable model described in equation (1), has $m - 1$ degrees of freedom for texture and $m - 1$ for shape. The expressiveness of the model can be increased by dividing faces into independent subregions that are morphed independently, for example into eyes, nose, mouth and a surrounding region (see Figure 2). Since all faces are assumed to be in correspondence, it is sufficient to define these regions on a reference face. This segmentation is equivalent to subdividing the vector space of faces into independent subspaces. A complete 3D face is generated by computing linear combinations for each segment separately and blending them at the borders according to an algorithm proposed for images by [7] .

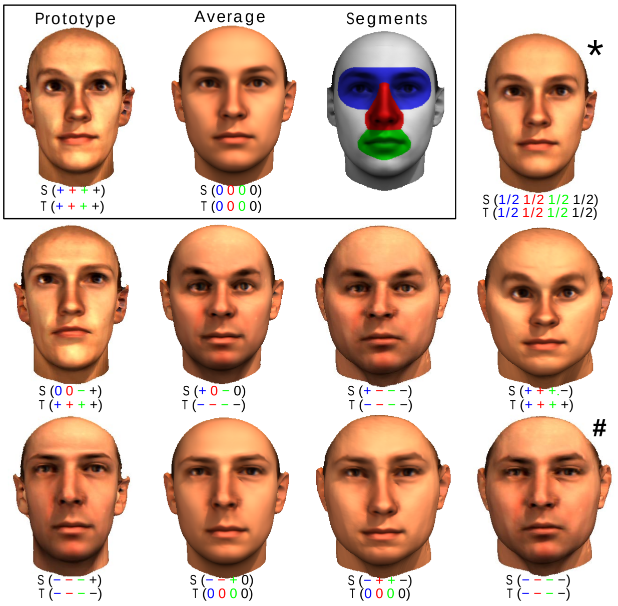


Figure 2: A single prototype adds a large variety of new faces to the morphable model. The deviation of a prototype from the average is added (+) or subtracted (-) from the average. A standard morph (*) is located halfway between average and the prototype. Subtracting the differences from the average yields an 'anti'-face (#). Adding and subtracting deviations independently for shape (S) and texture (T) on each of four segments produces a number of distinct faces.

3.1 Facial attributes

Shape and texture coefficients α_i and β_i in our morphable face model do not correspond to the facial attributes used in human language. While some facial attributes can easily be related to biophysical measurements [13, 10], such as the width of the mouth, others such as facial femininity or being more or less bony can hardly be described by numbers. In this section, we describe a method for mapping facial attributes, defined by a hand-labeled set of example faces, to the parameter space of our morphable model. At each position in face space (that is for any possible face), we define shape and texture vectors that, when added to or subtracted from a face, will manipulate a specific attribute while keeping all other attributes as constant as possible.

In a performance based technique [25], facial expressions can be transferred by recording two scans of the same individual with different expressions, and adding the differences $\Delta S = S_{expression} - S_{neutral}$, $\Delta T = T_{expression} - T_{neutral}$, to a different individual in a neutral expression.

Unlike facial expressions, attributes that are invariant for each individual are more difficult to isolate. The following method allows us to model facial attributes such as gender, fullness of faces, darkness of eyebrows, double chins, and hooked versus concave noses (Figure 3). Based on a set of faces (S_i, T_i) with manually assigned labels μ_i describing the markedness of the attribute, we compute

[1] Standard morphing between two faces ($m = 2$) is obtained if the parameters a_1, b_1 are varied between 0 and 1, setting $a_2 = 1 - a_1$ and $b_2 = 1 - b_1$.

[2] Due to the subtracted average vectors $\overline{S}$ and $\overline{T}$, the dimensions of $Span\{\Delta S_i\}$ and $Span\{\Delta T_i\}$ are at most $m - 1$.

weighted sums

$$\Delta S = \sum_{i=1}^{m} \mu_i (S_i - \overline{S}), \quad \Delta T = \sum_{i=1}^{m} \mu_i (T_i - \overline{T}). \qquad (3)$$

Multiples of $(\Delta S, \Delta T)$ can now be added to or subtracted from any individual face. For binary attributes, such as gender, we assign constant values μ_A for all m_A faces in class A, and $\mu_B \neq \mu_A$ for all m_B faces in B. Affecting only the scaling of ΔS and ΔT, the choice of μ_A, μ_B is arbitrary.

To justify this method, let $\mu(S,T)$ be the overall function describing the markedness of the attribute in a face (S,T). Since $\mu(S,T)$ is not available per se for all (S,T), the regression problem of estimating $\mu(S,T)$ from a sample set of labeled faces has to be solved. Our technique assumes that $\mu(S,T)$ is a linear function. Consequently, in order to achieve a change $\Delta\mu$ of the attribute, there is only a single optimal direction $(\Delta S, \Delta T)$ for the whole space of faces. It can be shown that Equation (3) defines the direction with minimal variance-normalized length $\|\Delta S\|_M^2 = \langle \Delta S, C_S^{-1} \Delta S \rangle$, $\|\Delta T\|_M^2 = \langle \Delta T, C_T^{-1} \Delta T \rangle$.

A different kind of facial attribute is its “distinctiveness”, which is commonly manipulated in caricatures. The automated production of caricatures has been possible for many years [6]. This technique can easily be extended from 2D images to our morphable face model. Individual faces are caricatured by increasing their distance from the average face. In our representation, shape and texture coefficients α_i, β_i are simply multiplied by a constant factor.

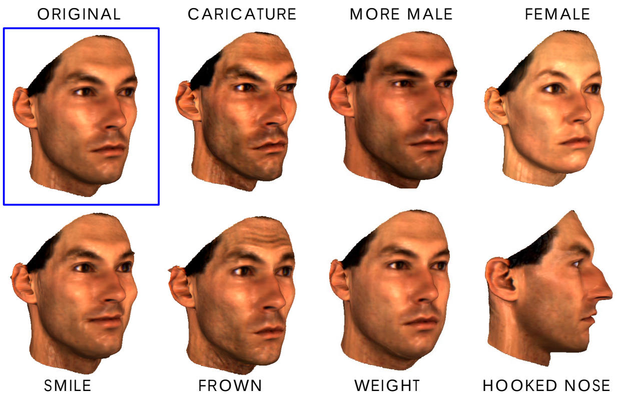


Figure 3: Variation of facial attributes of a single face. The appearance of an original face can be changed by adding or subtracting shape and texture vectors specific to the attribute.

4 Matching a morphable model to images

A crucial element of our framework is an algorithm for automatically matching the morphable face model to one or more images. Providing an estimate of the face’s 3D structure (Figure 4), it closes the gap between the specific manipulations described in Section 3.1, and the type of data available in typical applications.

Coefficients of the 3D model are optimized along with a set of rendering parameters such that they produce an image as close as possible to the input image. In an analysis-by-synthesis loop, the algorithm creates a texture mapped 3D face from the current model parameters, renders an image, and updates the parameters according to the residual difference. It starts with the average head and with rendering parameters roughly estimated by the user.

Model Parameters: Facial shape and texture are defined by coefficients α_j and β_j, $j = 1, ..., m-1$ (Equation 1). Rendering parameters $\vec{\rho}$ contain camera position (azimuth and elevation), object scale, image plane rotation and translation, intensity $i_{r,amb}, i_{g,amb}, i_{b,amb}$ of ambient light, and intensity $i_{r,dir}, i_{g,dir}, i_{b,dir}$ of directed light. In order to handle photographs taken under a wide variety of conditions, $\vec{\rho}$ also includes color contrast as well as offset and gain in the red, green, and blue channel. Other parameters, such as camera distance, light direction, and surface shininess, remain fixed to the values estimated by the user.

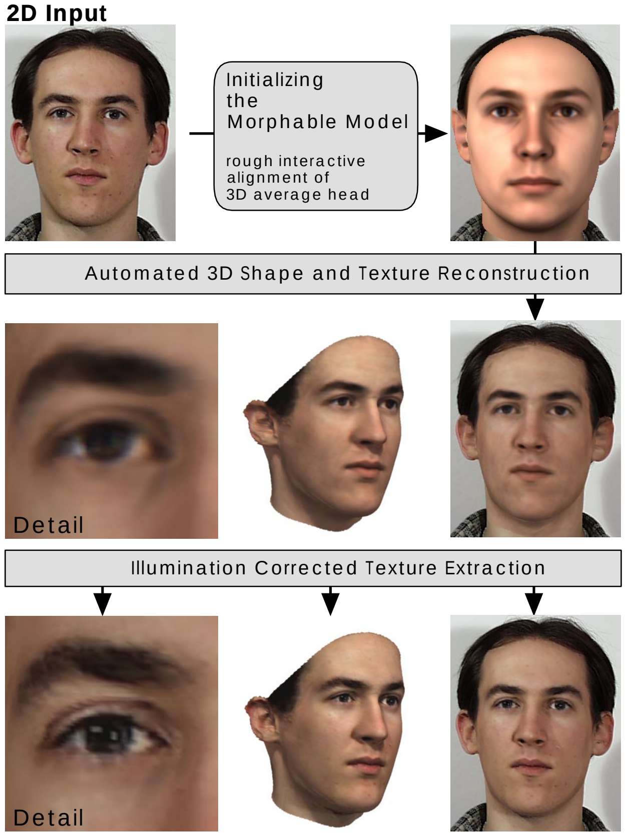

2D Input
Initializing the Morphable Model
rough interactive alignment of 3D average head
Automated 3D Shape and Texture Reconstruction
Detail
Illumination Corrected Texture Extraction
Detail

Figure 4: Processing steps for reconstructing 3D shape and texture of a new face from a single image. After a rough manual alignment of the average 3D head (top row), the automated matching procedure fits the 3D morphable model to the image (center row). In the right column, the model is rendered on top of the input image. Details in texture can be improved by illumination-corrected texture extraction from the input (bottom row).

From parameters $(\vec{\alpha}, \vec{\beta}, \vec{\rho})$, colored images

$$\mathbf{I}_{model}(x,y) = (I_{r,mod}(x,y), I_{g,mod}(x,y), I_{b,mod}(x,y))^T \quad (4)$$

are rendered using perspective projection and the Phong illumination model. The reconstructed image is supposed to be closest to the input image in terms of Euclidean distance

$$E_I = \textstyle\sum_{x,y} \|\mathbf{I}_{input}(x,y) - \mathbf{I}_{model}(x,y)\|^2.$$

Matching a 3D surface to a given image is an ill-posed problem. Along with the desired solution, many non-face-like surfaces lead to the same image. It is therefore essential to impose constraints on the set of solutions. In our morphable model, shape and texture vectors are restricted to the vector space spanned by the database.

Within the vector space of faces, solutions can be further restricted by a tradeoff between matching quality and prior probabilities, using $P(\vec{\alpha})$, $P(\vec{\beta})$ from Section 3 and an ad-hoc estimate of $P(\vec{\rho})$. In terms of Bayes decision theory, the problem is to find the set of parameters $(\vec{\alpha}, \vec{\beta}, \vec{\rho})$ with maximum posterior probability, given an image $\mathbf{I}_{input}$. While $\vec{\alpha}$, $\vec{\beta}$, and rendering parameters $\vec{\rho}$ completely determine the predicted image I_{model}, the observed image I_{input} may vary due to noise. For Gaussian noise with a standard deviation σ_N, the likelihood to observe I_{input} is $p(\mathbf{I}_{input}|\vec{\alpha}, \vec{\beta}, \vec{\rho}) \sim \exp[\frac{-1}{2\sigma_N^2} \cdot E_I]$. Maximum posterior probability is then achieved by minimizing the cost function

$$E = \frac{1}{\sigma_N^2} E_I + \sum_{j=1}^{m-1} \frac{\alpha_j^2}{\sigma_{S,j}^2} + \sum_{j=1}^{m-1} \frac{\beta_j^2}{\sigma_{T,j}^2} + \sum_j \frac{(\rho_j - \bar{\rho}_j)^2}{\sigma_{\rho,j}^2} \quad (5)$$

The optimization algorithm described below uses an estimate of E based on a random selection of surface points. Predicted color values $\mathbf{I}_{model}$ are easiest to evaluate in the centers of triangles. In the center of triangle k, texture $(\bar{R}_k, \bar{G}_k, \bar{B}_k)^T$ and 3D location $(\bar{X}_k, \bar{Y}_k, \bar{Z}_k)^T$ are averages of the values at the corners. Perspective projection maps these points to image locations $(\bar{p}_{x,k}, \bar{p}_{y,k})^T$. Surface normals $\mathbf{n}_k$ of each triangle k are determined by the 3D locations of the corners. According to Phong illumination, the color components $I_{r,model}$, $I_{g,model}$ and $I_{b,model}$ take the form

$$I_{r,model,k} = (i_{r,amb} + i_{r,dir} \cdot (\mathbf{n}_k \mathbf{l}))\bar{R}_k + i_{r,dir} s \cdot (\mathbf{r}_k \mathbf{v}_k)^\nu \quad (6)$$

where $\mathbf{l}$ is the direction of illumination, $\mathbf{v}_k$ the normalized difference of camera position and the position of the triangle's center, and $\mathbf{r}_k = 2(\mathbf{nl})\mathbf{n} - \mathbf{l}$ the direction of the reflected ray. s denotes surface shininess, and ν controls the angular distribution of the specular reflection. Equation (6) reduces to $I_{r,model,k} = i_{r,amb}\bar{R}_k$ if a shadow is cast on the center of the triangle, which is tested in a method described below.

For high resolution 3D meshes, variations in $\mathbf{I}_{model}$ across each triangle $k \in \{1, ..., n_t\}$ are small, so E_I may be approximated by

$$E_I \approx \sum_{k=1}^{n_t} a_k \cdot \|\mathbf{I}_{input}(\bar{p}_{x,k}, \bar{p}_{y,k}) - \mathbf{I}_{model,k}\|^2,$$

where a_k is the image area covered by triangle k. If the triangle is occluded, $a_k = 0$.

In gradient descent, contributions from different triangles of the mesh would be redundant. In each iteration, we therefore select a random subset $\mathcal{K} \subset \{1, ..., n_t\}$ of 40 triangles k and replace E_I by

$$E_{\mathcal{K}} = \sum_{k \in \mathcal{K}} \|\mathbf{I}_{input}(\bar{p}_{x,k}, \bar{p}_{y,k}) - \mathbf{I}_{model,k})\|^2. \quad (7)$$

The probability of selecting k is $p(k \in \mathcal{K}) \sim a_k$. This method of stochastic gradient descent [16] is not only more efficient computationally, but also helps to avoid local minima by adding noise to the gradient estimate.

Before the first iteration, and once every 1000 steps, the algorithm computes the full 3D shape of the current model, and 2D positions $(p_x, p_y)^T$ of all vertices. It then determines a_k, and detects hidden surfaces and cast shadows in a two-pass z-buffer technique. We assume that occlusions and cast shadows are constant during each subset of iterations.

Parameters are updated depending on analytical derivatives of the cost function E, using $\alpha_j \mapsto \alpha_j - \lambda_j \cdot \frac{\partial E}{\partial \alpha_j}$, and similarly for β_j and ρ_j, with suitable factors λ_j.

Derivatives of texture and shape (Equation 1) yield derivatives of 2D locations $(\bar{p}_{x,k}, \bar{p}_{y,k})^T$, surface normals $\mathbf{n}_k$, vectors $\mathbf{v}_k$ and $\mathbf{r}_k$, and $\mathbf{I}_{model,k}$ (Equation 6) using chain rule. From Equation (7), partial derivatives $\frac{\partial E_{\mathcal{K}}}{\partial \alpha_j}$, $\frac{\partial E_{\mathcal{K}}}{\partial \beta_j}$, and $\frac{\partial E_{\mathcal{K}}}{\partial \rho_j}$ can be obtained.

Coarse-to-Fine: In order to avoid local minima, the algorithm follows a coarse-to-fine strategy in several respects:

a) The first set of iterations is performed on a down-sampled version of the input image with a low resolution morphable model.

b) We start by optimizing only the first coefficients α_j and β_j controlling the first principal components, along with all parameters ρ_j. In subsequent iterations, more and more principal components are added.

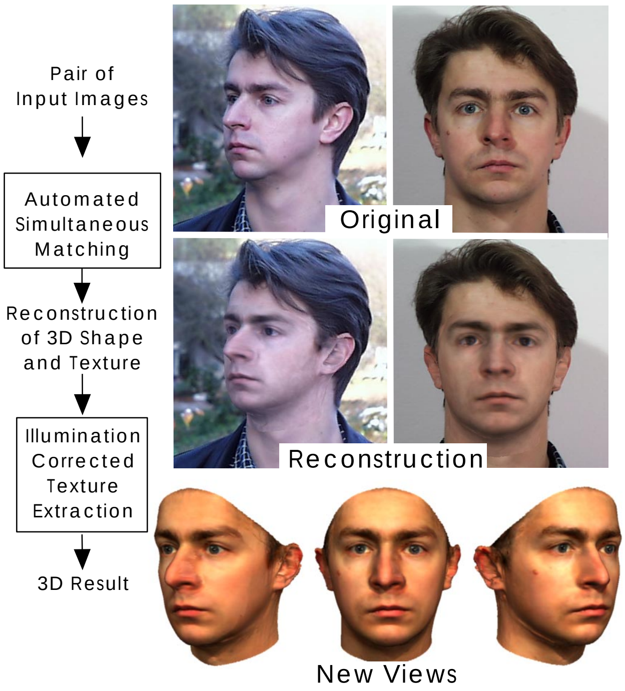

Pair of Input Images
Automated Simultaneous Matching
Reconstruction of 3D Shape and Texture
Illumination Corrected Texture Extraction
3D Result
Original
Reconstruction
New Views

Figure 5: Simultaneous reconstruction of 3D shape and texture of a new face from two images taken under different conditions. In the center row, the 3D face is rendered on top of the input images.

c) Starting with a relatively large σ_N, which puts a strong weight on prior probability in equation (5) and ties the optimum towards the prior expectation value, we later reduce σ_N to obtain maximum matching quality.

d) In the last iterations, the face model is broken down into segments (Section 3). With parameters ρ_j fixed, coefficients α_j and β_j are optimized independently for each segment. This increased number of degrees of freedom significantly improves facial details.

Multiple Images: It is straightforward to extend this technique to the case where several images of a person are available (Figure 5). While shape and texture are still described by a common set of α_j and β_j, there is now a separate set of ρ_j for each input image. E_I is replaced by a sum of image distances for each pair of input and model images, and all parameters are optimized simultaneously.

Illumination-Corrected Texture Extraction: Specific features of individual faces that are not captured by the morphable model, such as blemishes, are extracted from the image in a subsequent texture adaptation process. Extracting texture from images is a technique widely used in constructing 3D models from images (e.g. [28]). However, in order to be able to change pose and illumination, it is important to separate pure albedo at any given point from the influence of shading and cast shadows in the image. In our approach, this can be achieved because our matching procedure provides an estimate of 3D shape, pose, and illumination conditions. Subsequent to matching, we compare the prediction $\mathbf{I}_{mod,i}$ for each vertex i with $\mathbf{I}_{input}(p_{x,i}, p_{y,i})$, and compute the change in texture (R_i, G_i, B_i) that accounts for the difference. In areas occluded in the image, we rely on the prediction made by the model. Data from multiple images can be blended using methods similar to [28].

4.1 Matching a morphable model to 3D scans

The method described above can also be applied to register new 3D faces. Analogous to images, where perspective projection

$P : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ and an illumination model define a colored image $\mathbf{I}(x,y) = (R(x,y),G(x,y),B(x,y))^T$, laser scans provide a two-dimensional cylindrical parameterization of the surface by means of a mapping $C : \mathcal{R}^3 \rightarrow \mathcal{R}^2, \quad (x,y,z) \mapsto (h,\phi)$. Hence, a scan can be represented as

$$\mathbf{I}(h,\phi) = (R(h,\phi),G(h,\phi),B(h,\phi),r(h,\phi))^T. \qquad (8)$$

In a face (S,T), defined by shape and texture coefficients α_j and β_j (Equation 1), vertex i with texture values (R_i,G_i,B_i) and cylindrical coordinates (r_i,h_i,ϕ_i) is mapped to $\mathbf{I}_{model}(h_i,\phi_i) = (R_i,G_i,B_i,r_i)^T$. The matching algorithm from the previous section now determines α_j and β_j minimizing

$$E = \sum_{h,\phi} \|\mathbf{I}_{input}(h,\phi) - \mathbf{I}_{model}(h,\phi)\|^2.$$

5 Building a morphable model

In this section, we describe how to build the morphable model from a set of unregistered 3D prototypes, and to add a new face to the existing morphable model, increasing its dimensionality.

The key problem is to compute a dense point-to-point correspondence between the vertices of the faces. Since the method described in Section 4.1 finds the best match of a given face only within the range of the morphable model, it cannot add new dimensions to the vector space of faces. To determine residual deviations between a novel face and the best match within the model, as well as to set unregistered prototypes in correspondence, we use an optic flow algorithm that computes correspondence between two faces without the need of a morphable model [35]. The following section summarizes this technique.

5.1 3D Correspondence using Optic Flow

Initially designed to find corresponding points in grey-level images $I(x,y)$, a gradient-based optic flow algorithm [2] is modified to establish correspondence between a pair of 3D scans $\mathbf{I}(h,\phi)$ (Equation 8), taking into account color and radius values simultaneously [35]. The algorithm computes a flow field $(\delta h(h,\phi), \delta\phi(h,\phi))$ that minimizes differences of $\|\mathbf{I}_1(h,\phi) - \mathbf{I}_2(h+\delta h, \phi+\delta\phi)\|$ in a norm that weights variations in texture and shape equally. Surface properties from differential geometry, such as mean curvature, may be used as additional components in $\mathbf{I}(h,\phi)$.

On facial regions with little structure in texture and shape, such as forehead and cheeks, the results of the optic flow algorithm are sometimes spurious. We therefore perform a smooth interpolation based on simulated relaxation of a system of flow vectors that are coupled with their neighbors. The quadratic coupling potential is equal for all flow vectors. On high-contrast areas, components of flow vectors orthogonal to edges are bound to the result of the previous optic flow computation. The system is otherwise free to take on a smooth minimum-energy arrangement. Unlike simple filtering routines, our technique fully retains matching quality wherever the flow field is reliable. Optic flow and smooth interpolation are computed on several consecutive levels of resolution.

Constructing a morphable face model from a set of unregistered 3D scans requires the computation of the flow fields between each face and an arbitrary reference face. Given a definition of shape and texture vectors S_{ref} and T_{ref} for the reference face, S and T for each face in the database can be obtained by means of the point-to-point correspondence provided by $(\delta h(h,\phi), \delta\phi(h,\phi))$.

5.2 Bootstrapping the model

Because the optic flow algorithm does not incorporate any constraints on the set of solutions, it fails on some of the more unusual faces in the database. Therefore, we modified a bootstrapping algorithm to iteratively improve correspondence, a method that has been used previously to build linear image models [36].

The basic recursive step: Suppose that an existing morphable model is not powerful enough to match a new face and thereby find correspondence with it. The idea is first to find rough correspondences to the novel face using the (inadequate) morphable model and then to improve these correspondences by using an optic flow algorithm.

Starting from an arbitrary face as the temporary reference, preliminary correspondence between all other faces and this reference is computed using the optic flow algorithm. On the basis of these correspondences, shape and texture vectors S and T can be computed. Their average serves as a new reference face. The first morphable model is then formed by the most significant components as provided by a standard PCA decomposition. The current morphable model is now matched to each of the 3D faces according to the method described in Section 4.1. Then, the optic flow algorithm computes correspondence between the 3D face and the approximation provided by the morphable model. Combined with the correspondence implied by the matched model, this defines a new correspondence between the reference face and the example.

Iterating this procedure with increasing expressive power of the model (by increasing the number of principal components) leads to reliable correspondences between the reference face and the examples, and finally to a complete morphable face model.

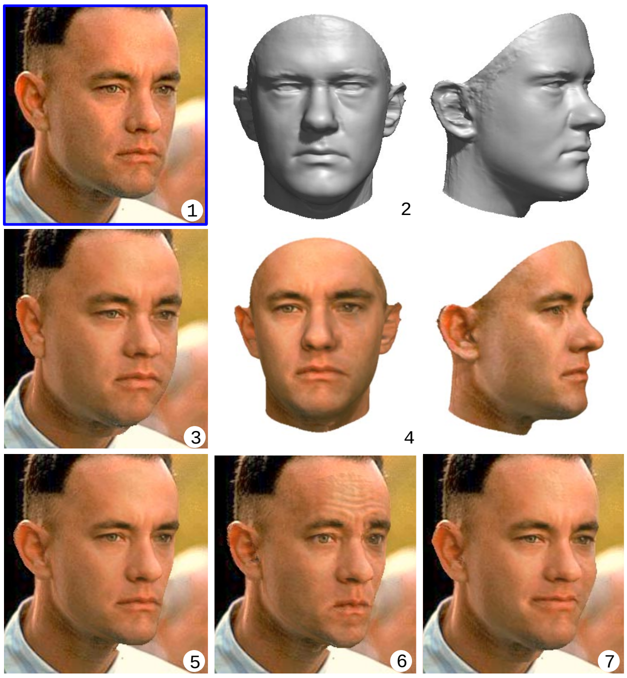


Figure 6: Matching a morphable model to a single image (1) of a face results in a 3D shape (2) and a texture map estimate. The texture estimate can be improved by additional texture extraction (4). The 3D model is rendered back into the image after changing facial attributes, such as gaining (3) and loosing weight (5), frowning (6), or being forced to smile (7).

6 Results

We built a morphable face model by automatically establishing correspondence between all of our 200 exemplar faces. Our interactive

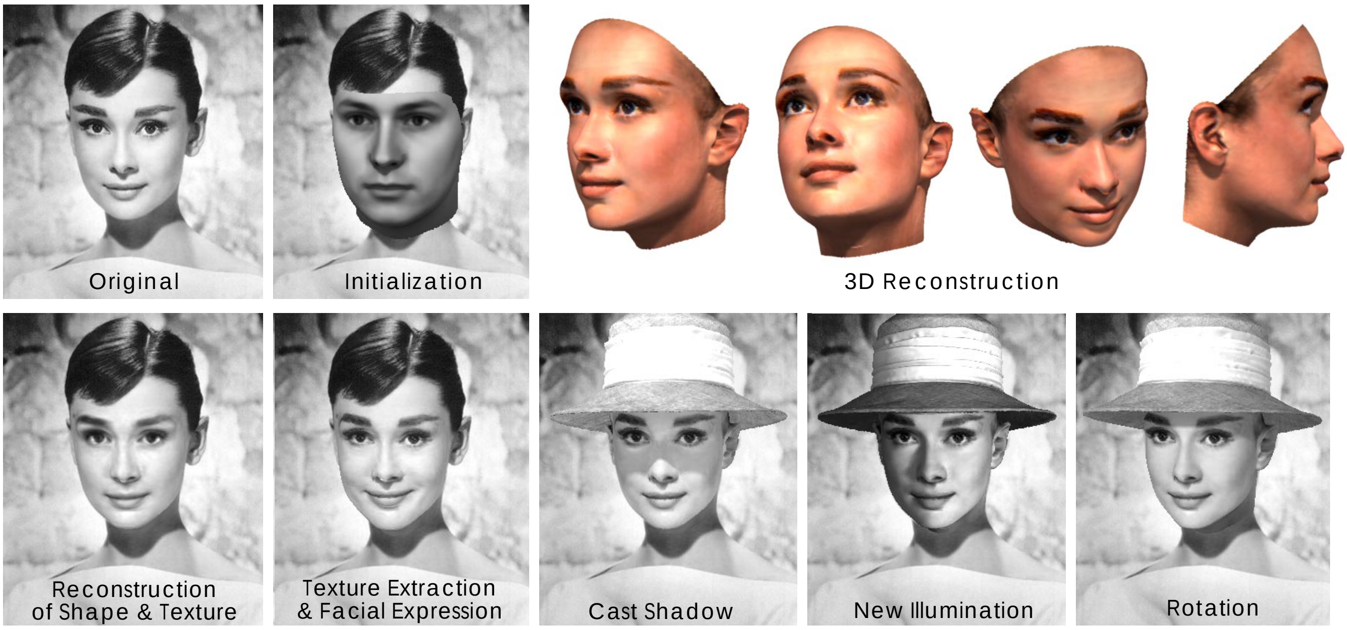


Figure 7: After manual initialization, the algorithm automatically matches a colored morphable model (color contrast set to zero) to the image. Rendering the inner part of the 3D face on top of the image, new shadows, facial expressions and poses can be generated.

face modeling system enables human users to create new characters and to modify facial attributes by varying the model coefficients. Within the constraints imposed by prior probability, there is a large variability of possible faces, and all linear combinations of the exemplar faces look natural.

We tested the expressive power of our morphable model by automatically reconstructing 3D faces from photographs of arbitrary Caucasian faces of middle age that were not in the database. The images were either taken by us using a digital camera (Figures 4, 5), or taken under arbitrary unknown conditions (Figures 6, 7).

In all examples, we matched a morphable model built from the first 100 shape and the first 100 texture principal components that were derived from the whole dataset of 200 faces. Each component was additionally segmented in 4 parts (see Figure 2). The whole matching procedure was performed in 10^5 iterations. On an SGI R10000 processor, computation time was 50 minutes.

Reconstructing the true 3D shape and texture of a face from a single image is an ill-posed problem. However, to human observers who also know only the input image, the results obtained with our method look correct. When compared with a real image of the rotated face, differences usually become only visible for large rotations of more than $60°$.

There is a wide variety of applications for 3D face reconstruction from 2D images. As demonstrated in Figures 6 and 7, the results can be used for automatic post-processing of a face within the original picture or movie sequence.

Knowing the 3D shape of a face in an image provides a segmentation of the image into face area and background. The face can be combined with other 3D graphic objects, such as glasses or hats, and then be rendered in front of the background, computing cast shadows or new illumination conditions (Fig. 7). Furthermore, we can change the appearance of the face by adding or subtracting specific attributes. If previously unseen backgrounds become visible, we fill the holes with neighboring background pixels (Fig. 6).

We also applied the method to paintings such as Leonardo's Mona Lisa (Figure 8). Due to unusual (maybe unrealistic) lighting, illumination-corrected texture extraction is difficult here. We therefore apply a different method for transferring all details of the painting to novel views. For new illumination, we render two images of the reconstructed 3D face with different illumination, and multiply relative changes in pixel values (Figure 8, bottom left) by the original values in the painting (bottom center). For a new pose (bottom right), differences in shading are transferred in a similar way, and the painting is then warped according to the 2D projections of 3D vertex displacements of the reconstructed shape.

7 Future work

Issues of implementation: We plan to speed up our matching algorithm by implementing a simplified Newton-method for minimizing the cost function (Equation 5). Instead of the time consuming computation of derivatives for each iteration step, a global mapping of the matching error into parameter space can be used [9].

Data reduction applied to shape and texture data will reduce redundancy of our representation, saving additional computation time.

Extending the database: While the current database is sufficient to model Caucasian faces of middle age, we would like to extend it to children, to elderly people as well as to other races.

We also plan to incorporate additional 3D face examples representing the time course of facial expressions and visemes, the face variations during speech.

The laser scanning technology we used, unfortunately, does not allow us to collect dynamical 3D face data, as each scanning cycle takes at least 10 seconds. Consequently, our current example set of facial expressions is restricted to those that can be kept static by the scanned subjects. However, the development of fast optical 3D digitizers [27] will allow us to apply our method to streams of 3D data during speech and facial expressions.

Extending the face model: Our current morphable model is restricted to the face area, because a sufficient 3D model of hair cannot be obtained with our laser scanner. For animation, the missing part of the head can be automatically replaced by a standard hair style or a hat, or by hair that is modeled using interactive manual segmentation and adaptation to a 3D model [30, 28]. Automated reconstruction of hair styles from images is one of the future challenges.

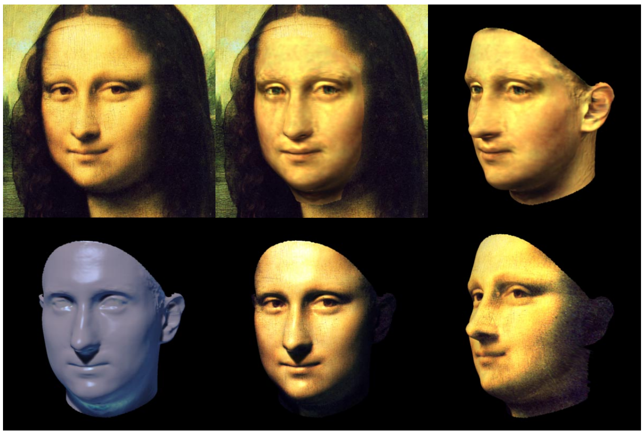


Figure 8: Reconstructed 3D face of Mona Lisa (top center and right). For modifying the illumination, relative changes in color (bottom left) are computed on the 3D face, and then multiplied by the color values in the painting (bottom center). Additional warping generates new orientations (bottom right, see text), while details of the painting, such as brush strokes or cracks, are retained.

8 Acknowledgment

We thank Michael Langer, Alice O'Toole, Tomaso Poggio, Heinrich Bülthoff and Wolfgang Straßer for reading the manuscript and for many insightful and constructive comments. In particular, we thank Marney Smyth and Alice O'Toole for their perseverance in helping us to obtain the following. **Photo Credits:** Original image in Fig. 6: Courtesy of Paramount/VIACOM. Original image in Fig. 7: MPTV/interTOPICS.

References

[1] T. Akimoto, Y. Suenaga, and R.S. Wallace. Automatic creation of 3D facial models. *IEEE Computer Graphics and Applications*, 13(3):16–22, 1993.

[2] J.R. Bergen and R. Hingorani. Hierarchical motion-based frame rate conversion. Technical report, David Sarnoff Research Center Princeton NJ 08540, 1990.

[3] P. Bergeron and P. Lachapelle. Controlling facial expressions and body movements. In *Advanced Computer Animation, SIGGRAPH '85 Tutorials*, volume 2, pages 61–79, New York, 1985. ACM.

[4] D. Beymer and T. Poggio. Image representation for visual learning. *Science*, 272:1905–1909, 1996.

[5] D. Beymer, A. Shashua, and T. Poggio. Example-based image analysis and synthesis. A.I. Memo No. 1431, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1993.

[6] S. E. Brennan. The caricature generator. *Leonardo*, 18:170–178, 1985.

[7] P.J. Burt and E.H. Adelson. Merging images through pattern decomposition. In *Applications of Digital Image Processing VIII*, number 575, pages 173–181. SPIE The International Society for Optical Engeneering, 1985.

[8] C.S. Choi, T. Okazaki, H. Harashima, and T. Takebe. A system of analyzing and synthesizing facial images. In *Proc. IEEE Int. Symposium of Circuit and Syatems (ISCAS91)*, pages 2665–2668, 1991.

[9] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In Burkhardt and Neumann, editors, *Computer Vision – ECCV'98 Vol. II*, Freiburg, Germany, 1998. Springer, Lecture Notes in Computer Science 1407.

[10] D. DeCarlos, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 67–74, 1998.

[11] S. DiPaola. Extending the range of facial types. *Journal of Visualization and Computer Animation*, 2(4):129–131, 1991.

[12] G.J. Edwards, A. Lanitis, C.J. Taylor, and T.F. Cootes. Modelling the variability in face images. In *Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition*, IEEE Comp. Soc. Press, Los Alamitos, CA, 1996.

[13] L.G. Farkas. *Anthropometry of the Head and Face*. RavenPress, New York, 1994.

[14] B. Guenter, C. Grimm, D. Wolf, H. Malvar, and F. Pighin. Making faces. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 55–66, 1998.

[15] I.T. Jollife. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[16] M. Jones and T. Poggio. Multidimensional morphable models: A framework for representing and matching object classes. In *Proceedings of the Sixth International Conference on Computer Vision*, Bombay, India, 1998.

[17] R. M. Koch, M. H. Gross, and A. A. Bosshard. Emotion editing using finite elements. In *Proceedings of the Eurographics '98, COMPUTER GRAPHICS Forum, Vol. 17, No. 3*, pages C295–C302, Lisbon, Portugal, 1998.

[18] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.

[19] Y.C. Lee, D. Terzopoulos, and Keith Waters. Constructing physics-based facial models of individuals. *Visual Computer*, Proceedings of Graphics Interface '93:1–8, 1993.

[20] Y.C. Lee, D. Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *SIGGRAPH '95 Conference Proceedings*, pages 55–62, Los Angels, 1995. ACM.

[21] J. P. Lewis. Algorithms for solid noise synthesis. In *SIGGRAPH '89 Conference Proceedings*, pages 263–270. ACM, 1989.

[22] N. Magneneat-Thalmann, H. Minh, M. Angelis, and D. Thalmann. Design, transformation and animation of human faces. *Visual Computer*, 5:32–39, 1989.

[23] L. Moccozet and N. Magnenat-Thalmann. Dirichlet free-form deformation and their application to hand simulation. In *Computer Animation'97*, 1997.

[24] F. I. Parke and K. Waters. *Computer Facial Animation*. AKPeters, Wellesley, Massachusetts, 1996.

[25] F.I. Parke. Computer generated animation of faces. In *ACM National Conference*. ACM, November 1972.

[26] F.I. Parke. *A Parametric Model of Human Faces*. PhD thesis, University of Utah, Salt Lake City, 1974.

[27] M. Petrow, A. Talapov, T. Robertson, A. Lebedev, A. Zhilyaev, and L. Polonskiy. Optical 3D digitizer: Bringing life to virtual world. *IEEE Computer Graphics and Applications*, 18(3):28–37, 1998.

[28] F. Pighin, J. Hecker, D. Lischinski, Szeliski R, and D. Salesin. Synthesizing realistic facial expressions from photographs. In *Computer Graphics Proceedings SIGGRAPH'98*, pages 75–84, 1998.

[29] S. Platt and N. Badler. Animating facial expression. *Computer Graphics*, 15(3):245–252, 1981.

[30] G. Sannier and N. Magnenat-Thalmann. A user-friendly texture-fitting methodology for virtual humans. In *Computer Graphics International'97*, 1997.

[31] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4:519–554, 1987.

[32] D. Terzopoulos and Keith Waters. Physically-based facial modeling, analysis, and animation. *Visualization and Computer Animation*, 1:73–80, 1990.

[33] Demetri Terzopoulos and Hong Qin. Dynamic NURBS with geometric constraints to interactive sculpting. *ACM Transactions on Graphics*, 13(2):103–136, April 1994.

[34] J. T. Todd, S. M. Leonard, R. E. Shaw, and J. B. Pittenger. The perception of human growth. *Scientific American*, 1242:106–114, 1980.

[35] T. Vetter and V. Blanz. Estimating coloured 3d face models from single images: An example based approach. In Burkhardt and Neumann, editors, *Computer Vision – ECCV'98 Vol. II*, Freiburg, Germany, 1998. Springer, Lecture Notes in Computer Science 1407.

[36] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition – CVPR'97*, Puerto Rico, USA, 1997. IEEE Computer Society Press.

[37] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

[38] Keith Waters. A muscle model for animating three-dimensional facial expression. *Computer Graphics*, 22(4):17–24, 1987.

EUROGRAPHICS 2003

Forensic Applications

Jörg Haber

Typical Applications

- police work:
  - growth simulation / aging of missing children
  - facial reconstruction from skeletal remains
- important for tracing and identification
- based on anthropometric data

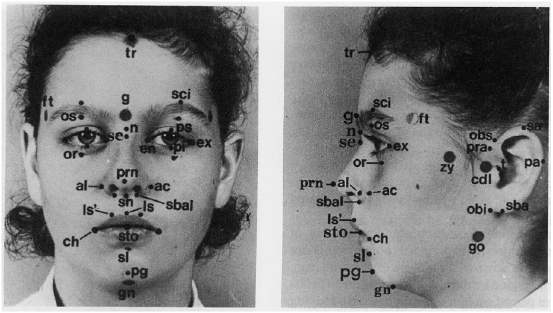
Anthropometric Data

Data collected over decades

- facial measurements: landmarks
- populations vary by:
  - ethnicity (caucasian, asian, …)
  - age (1-25 yrs. for growth measurements)
  - sex
- measurements consist of:
  - distances: axis-aligned, euclidean, arc-length
  - angles
  - proportions

Anthropometric Data

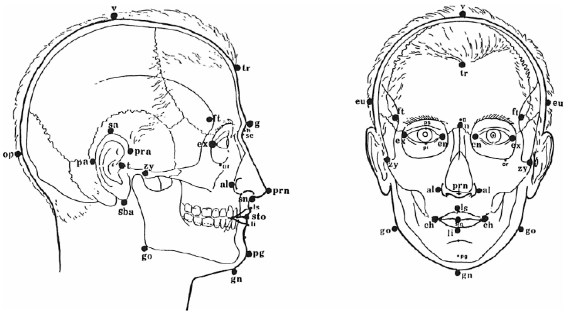
Examples for measurements:



Images: Farkas: "Anthropometry of the Head and Face", 1994

Anthropometric Data

Examples for measurements:

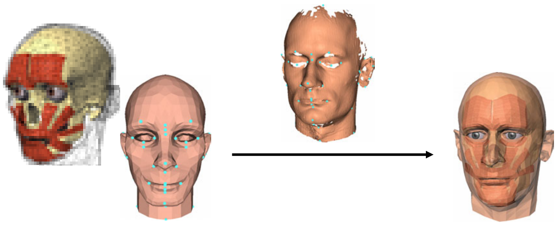


Images: Farkas: "Anthropometry of the Head and Face", 1994

Deformable Head Model

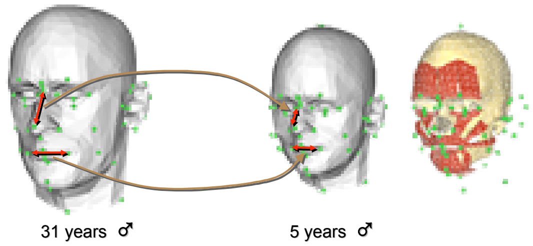
Idea: use landmarks for head deformation

- structured, animatable reference head model
- tagged with landmarks
- thin-plate spline interpolation for deformation



Growth Simulation

- analyze landmark positions of given age
- compute new landmarks for target age
- deform reference head model to fit new landmarks



Growth Simulation

Derive new measurements for age change

- given input head model, age, sex, ethnicity
- examine landmarks on input model:
  - find deviation from statistical data
- look up statistics for target age using same deviation
- compute new landmarks:
  - best fit for target age measurements

Growth Simulation

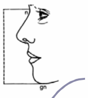
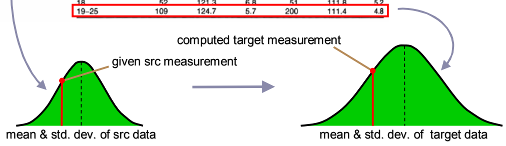


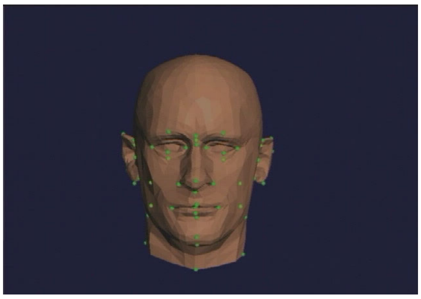
TABLE A–II–4. *Morphological height of the face (n-gn) (mm)*

Age (years)	Male *N*	Male Mean	Male SD	Female *N*	Female Mean	Female SD
0–5 months	8	70.0	4.5	5	68.0	4.5
6–12 months	20	70.5	4.8	8	72.7	4.4
1	18	80.6	4.8	20	77.2	4.9
2	31	87.5	3.5	31	83.8	3.9
3	30	88.5	3.5	30	86.9	2.4
4	30	96.4	4.3	30	92.6	3.6
5	30	96.7	3.5	30	96.5	4.6
6	50	98.5	5.0	50	95.7	4.4
7	50	99.5	5.0	50	98.3	3.7
8	51	101.8	4.9	51	98.1	5.4
9	51	102.7	5.3	50	101.3	5.3
10	50	105.2	4.5	49	103.9	5.0
11	50	107.1	6.0	51	104.7	5.0
12	52	109.1	5.4	53	108.2	4.6
13	50	111.6	5.7	49	109.1	5.0
14	49	114.1	6.5	51	110.7	5.3
15	50	119.1	5.7	51	111.0	5.1
16	50	120.9	4.6	51	113.5	6.0
17	49	120.9	7.1	51	112.0	4.7
18	52	121.3	6.8	51	111.8	5.2
19–25	109	124.7	5.7	200	111.4	4.8

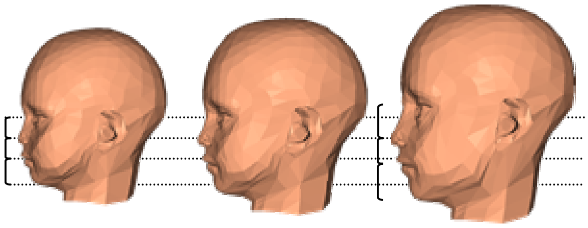
Table: Farkas: "Anthropometry of the Head and Face", 1994


computed target measurement
given src measurement
mean & std. dev. of src data
mean & std. dev. of target data

Growth Simulation Movie



Growth Simulation Examples



age 1 year

age 5 years (original)

age 20 years

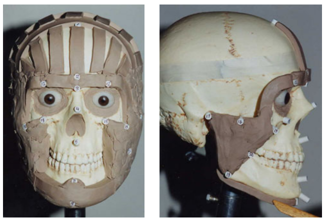
Limitations

It´s only statistics!

- landmarks are sparsely distributed
  - lots of source characteristics are maintained
- positioning in normal distribution valid?
  - does a child with a big nose have a big nose as an adult?
- accuracy depends on physical measurements taken decades ago
  - could be improved using 3D scanning
  - build up a big database of measurements?

Reconstruction of Faces

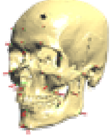
- traditional clay sculpting approach:
  - place tissue depth markers on the skull; length of pegs corresponds to anthropometric data
  - face is modeled using clay (→ artistic licence)



Images: Taylor: "Forensic Art and Illustration", 2001

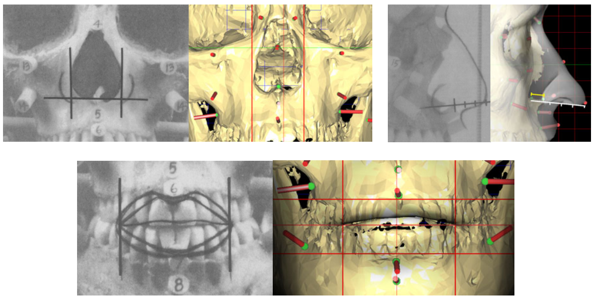
CG Approach

1. Acquisition of skull data (3D range scan, computer tomography)
2. interactive placement of landmarks on the virtual skull; tissue depth values assigned automatically from anthropometric data tables
3. automatic fitting of the reference head model to the prescribed skin surface positions ⇒ instantly animatable head model

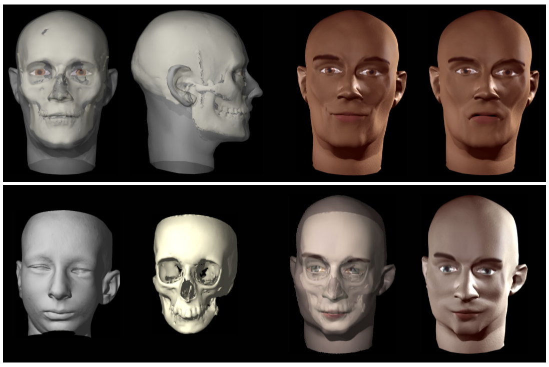


Additional Reconstruction Hints

- forensic art: many "rules of the thumb" to locate certain features of the face based on the skull shape



Results



Discussion

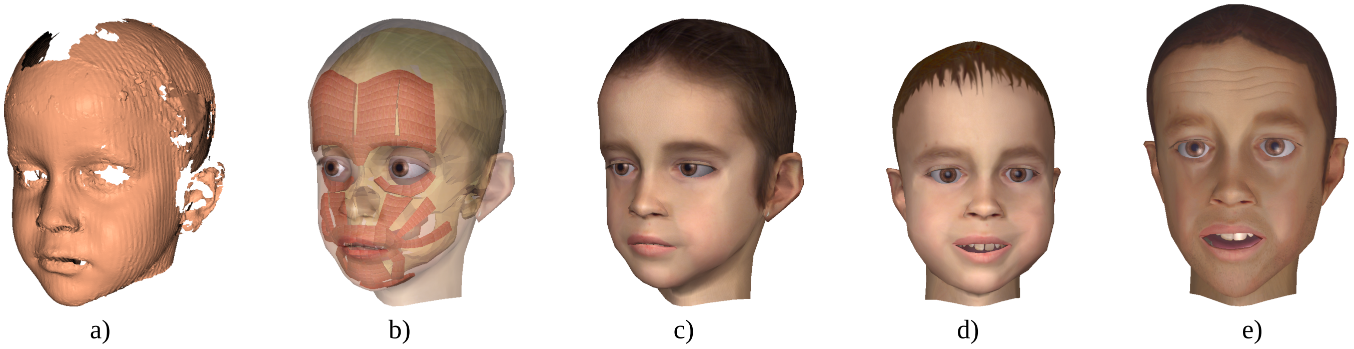
Again: it´s only statistics!

- our method mirrors the manual tissue depth method ⇒ same prediction power
- results show plausible reproduction of facial shape and proportions + some surprisingly well-matched details!
- advantages: very fast (a few hours instead of weeks), does not damage original skull
- need additional editing tools for hair, beards, wrinkles
- most promising: gather lots of data through simulation and evaluation; update tissue thickness tables with these data

Head shop: Generating animated head models with anatomical structure

Kolja Kähler Jörg Haber Hitoshi Yamauchi Hans-Peter Seidel

Max-Planck-Institut für Informatik, Saarbrücken



Head models generated from a range scan of a five year old boy: a) scan data; b) adapted, animatable head structure; c) textured; d) age changed to one year, smiling expression; e) age changed to 20 years, surprised expression.

Abstract

We present a versatile construction and deformation method for head models with anatomical structure, suitable for real-time physics-based facial animation. The model is equipped with landmark data on skin and skull, which allows us to deform the head in anthropometrically meaningful ways. On any deformed model, the underlying muscle and bone structure is adapted as well, such that the model remains completely animatable using the same muscle contraction parameters. We employ this general technique to fit a generic head model to imperfect scan data, and to simulate head growth from early childhood to adult age.

CR Categories: G.1.2 [Numerical Analysis]: Approximation—*approximation of surfaces*, *least squares approximation*; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*hierarchy and geometric transformations*, *physically based modeling*; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*animation*.

Keywords: Biological Modeling, Deformations, Facial Animation, Geometric Modeling, Morphing, Physically Based Animation

1 Introduction

Recent advances in facial animation systems show the potential of physics-based approaches, where the anatomical structure of a human head is simulated, including skin, muscles, and skull [Waters and Frisbie 1995; Lee et al. 1995]. On current hardware, this kind of techniques can be used to create detailed, realistic animations in real-time. As the model becomes more complex, the assembly of the components becomes more complicated, thus giving rise to a strong interest in automated methods for head model construction. Once a model exists, it is often desirable to change some of its characteristics, or generally adapt it to another head geometry, while retaining full animation capabilities.

The more realistically the heads of existing human individuals can be reproduced in the computer, the more appealing is the use of anthropometric methods and data for analysis and modification of head geometry [DeCarlo et al. 1998]. Measurements of human heads from different samples of the world population have been systematically collected over the past decades [Farkas 1994], resulting in a database of the predominant facial characteristics for individuals of different sex, age, and ethnicity. Anthropometric methods are usually based on *landmarks*, i.e. well-defined features on the face and – in the forensic sciences – also on the skull. In recent years, the topic of deformation of biological shapes described by such sets of landmark data has been treated by the emerging science of morphometrics [Bookstein 1997a].

We propose the use of anthropometric landmarks and an associated deformation technique based on thin-plate splines, arriving at a unified, elegant framework for a variety of tasks in facial modeling and animation. The added layer of abstraction over the implementation details of the structured head model allows for modification of the head geometry in terms of distance relations between facial features. Given a reference head model tagged with landmarks on the skin and bone layers, we automatically deform not only the outer skin geometry, but also the internal structure composed of muscles and skull. All head models derived from the reference head share the same set of animation parameters, i.e. muscles, enabling re-use of existing animation scripts.

The main contributions presented in this paper are:

- a general method to deform an animated head model with underlying anatomical structure, tagged with anthropometrically meaningful landmarks. The head model is suitable for real-time animation based on simulation of facial muscles and elastic skin properties.
- an algorithm to fit such a reference head model to even very poor scan data, using this deformation technique.
- a technique that utilizes the anthropometric measurements on the head model to simulate growth of a human head, employing the same deformation method, and resulting in animatable head models of an individual at different ages.

While we make extensive use of the body of knowledge and data collected for human faces, the general approach is applicable just as well to other classes of animated virtual creatures, provided a reference model with skull, muscles, and skin can be built.

appeared originally in Proc. ACM Symposium on Computer Animation 2002, pp. 55–64

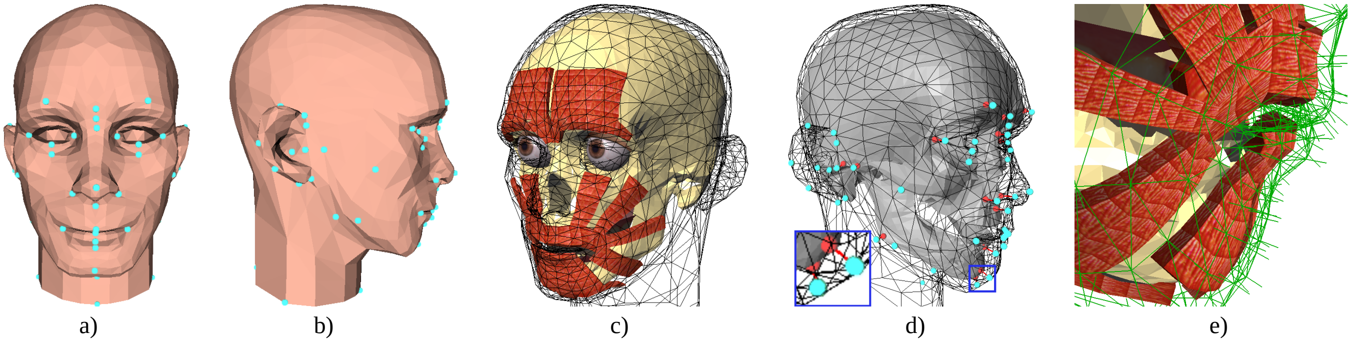


Figure 1: The reference head: a) head geometry with landmarks, front view; b) side view; c) skull and facial components; d) skull landmarks related to subset of skin landmarks; e) facial detail showing spring mesh connecting skin and muscles.

2 Previous and related work

Several facial animation systems use some approximation of layered anatomical structure. The idea of representing skin and muscles as separate entities was used by WATERS [1987], where muscle vectors and radial functions derived from linear and sphincter muscles specify deformations on a skin mesh. Also building on the idea of virtual muscles, physics-based approaches attempt to model the influence of muscle contraction onto the skin surface by approximating the biomechanical properties of skin. Typically, mass-spring or finite element networks are used for numerical simulation [Platt and Badler 1981; Lee et al. 1993; Koch et al. 1998]. TERZOPOULOS and WATERS [1990] automatically construct a layered model of the human face from an initial triangle mesh. The structure consists of three layers representing the muscle layer, dermis, and epidermis. The skull is approximated as an offset surface from the skin. This model was later simplified by LEE *et al.* [1995] for efficiency. Free-form deformations have been employed by CHADWICK *et al.* [1989] to shape the skin in a multi-layer model, which contains bones, muscles, fat tissue, and skin. SCHEEPERS *et al.* [1997] and WILHELMS and VAN GELDER [1997] introduced anatomy-based muscle models for animating humans and animals, focusing on the skeletal musculature. Skin tissue is represented only by an implicit surface with zero thickness [Wilhelms and Van Gelder 1997].

A variety of techniques exist to create a face model from images or scan data. In the method presented by LEE *et al.* [1995], animatable head models are constructed semi-automatically from range scans. A generic face mesh with embedded muscle vectors is adapted to range scans of human heads. This process relies on a planar parameterization of the range scans as delivered e.g. by the Cyberware digitizers. PIGHIN *et al.* [1998] interactively mark corresponding facial features in several photographs of an individual to deform a generic head model using radial basis functions. Animation is possible by capturing facial expressions in the process and blending between them. Employing a large database of several hundred scanned faces, BLANZ *et al.* [1999] are able to create a geometric head model from only a single photograph. The model has the same resolution as the range scans in the database and cannot be readily animated. CARR *et al.* [2001] also use radial basis functions to generate consistent meshes from incomplete scan data. In medical imaging, SZELISKI *et al.* [1996] minimize the distance between two surfaces obtained from volume scans of human heads by applying local free-form deformations [Sederberg and Parry 1986] and global polynomial deformations. The method does not require specification of corresponding features on the geometries.

A variational approach is presented by DECARLO *et al.* [1998] to create a range of static face models with realistic proportions. They use anthropometric measurements, which constrain the deformation of a generic head model represented by a B-spline surface.

Recently, the transfer of animations between different head models on the geometric level has been proposed [Noh and Neumann 2001]. Surface correspondences are obtained by specification of corresponding point pairs on the models. Heuristics for automatic feature detection are presented which help to automate the process.

The work on aging in human faces has so far concentrated on the appearance of the skin, neglecting the considerable geometric changes that occur during growth. WU *et al.* [1999; 1994] focus on generation of expressive wrinkles and skin aging effects. Their muscle-driven face model incorporates viscoelastic properties of skin. LEE *et al.* [1999] reconstruct textured low polygon face models from photographs of the members of a family, simulating age changes by blending geometry and textures between young and old family members. Wrinkle patterns are generated semi-automatically by considering muscle fiber orientation and feature points on the face. LANITIS *et al.* [1999] present a statistical face model to isolate age variations in face images for age estimation and aging simulation. TIDDEMAN *et al.* [2001] use wavelet-based methods to identify salient features such as age wrinkles in prototype facial images, and apply them to other images to change the apparent age.

3 The reference head model

Our system builds on a prototype head model that has been designed for use in our physics-based animation system. The model encapsulates five major structural components, shown in Figure 1:

- a triangle mesh for the *skin surface*. The edges are aligned to facial features to reduce animation artifacts. The tessellation is adjusted to the deformability of the facial regions.
- a layer of *virtual muscles* to control the animation. The muscles consist of arrays of fibers which can contract in a linear or circular fashion. We have modeled 24 of the major muscles responsible for facial expressions and speech articulation.
- an embedded *skull*, including a rotatable mandible, to which skin and muscles attach. The skull is also represented as a triangle mesh and is only used during initialization of the structure, not during animation itself.
- a *mass-spring system* connecting skin, muscles, and skull. Basically, the edges and vertices of the skin surface mesh are converted to springs and point masses, respectively. More springs are added to connect to underlying components and to preserve skin tissue volume during animation.
- separately modeled components for *eyes, teeth, and tongue*.

The model structure has been designed manually, employing the muscle editing methods presented in [Kähler et al. 2001]. We have enhanced the described muscle model in several respects. The ellipsoid segments have been replaced by a piecewise linear representation, to give the muscles a smoother, closed surface. As in [Kähler et al. 2001], skin vertices attach to muscles, and muscles in turn can be attached to the mandible. Since we derive all head models by deformation of the same reference head, we keep those assignments fixed. This makes expressions more reproducible between different head models, because instabilities in vertex re-attachment on the deformed models are avoided. To incorporate elastic properties of muscle fibers, the interaction of merged muscles is modeled more faithfully by using another simple spring mesh. The springs run along the middle axis of each muscle, connecting merged muscles by a common node. Finally, the *orbicularis oris* is divided into two muscles for upper and lower lip, and the center of contraction for each of these can be translated along a line. These added degrees of freedom allow more accurate protrusion and retraction of the lips, which is for example useful in speech animation.

The model is tagged with landmarks, defined on the skin and skull surfaces. We use these landmarks to control deformation of the head structure in our algorithms. The landmarks follow the conventions laid out in [Farkas 1994], where we have chosen a minimum subset of landmarks according to their prominence in the face and existence of a correspondence between skin and skull. There are in general far less landmarks on the skull than on the skin, since not every feature on the skin surface corresponds to one on the skull, cf. Figure 1 d). In our current model, we use 60 skin landmarks and 22 skull landmarks.

4 Animation and rendering overview

The focus of this paper is on the structured head model described in the previous section, and geometric methods of deformation of this model. To establish the context, we nonetheless need to describe the embedding of the model into the complete physics-based facial animation system. This section gives a necessarily brief overview of the issues involved, omitting most of the technical detail.

4.1 Animation control

Facial motion is controlled mainly by specifying muscle contractions over time. We explicitly specify these parameters for a number of keyframes, assembling facial expressions. For animation, we perform interpolation between these contraction values. The complex dynamic of the human face in motion is hard to reproduce in this manner, requiring higher level animation facilities that are beyond the scope of this paper. We have successfully used the muscle-based approach for the automatic generation of speech animation [Albrecht et al. 2002].

4.2 Physics-based simulation

The animation is controlled on the lowest level by muscle parameters from an animation script or from user interaction. During the simulation, the equations of motion for the mass-spring system are numerically integrated through time using a Verlet leapfrog integration scheme [Vesely 1994; Turner and Gobbetti 1998]. This explicit forward integration scheme provides better stability than the popular Euler method [Waters and Frisbie 1995; Lee et al. 1995] with similar ease of implementation.

The spring mesh structure is similar to [Kähler et al. 2001], which is advantageous for real-time animation: the complexity is relatively low compared to other layered approaches [Terzopoulos and Waters 1990; Lee et al. 1995]. Volume preservation and skull non-penetration constraints are integrated into the global solution of the equations of motion, obviating the need for local geometric criteria. Since the number of springs in the system is proportional to the number of edges and vertices of the head model, we chose to have a rather low resolution mesh to enable fast simulation updates.

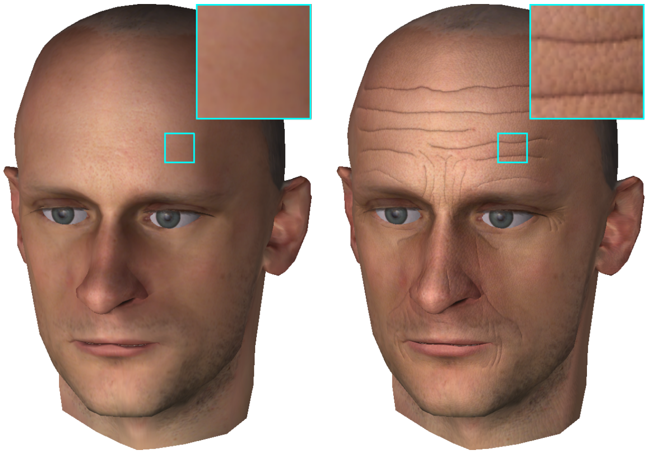


Figure 2: Comparison of an individual with simulated age wrinkles using plain OpenGL rendering (left) and our real-time skin shading algorithm (right).

4.3 Multi-threaded simulation and rendering

An important application in our system is speech animation, which requires very fast simulation updates in the range of 40 fps for real-time animation. While we can achieve these rates with our current model, the graphics performance of modern hardware allows for much higher rendering frame rates. Our facial animation system thus decouples simulation and rendering, exploiting dual processor systems by using individual threads for the physics-based simulation and the rendering of an animation [Haber et al. 2001].

4.4 Skin shading

To improve the visual appearance of the skin surface in real-time rendering, we employ the vertex program and register combiner features of the NVidia GeForce3 graphics board. In particular, we apply multitexturing using four texture units. The texture units are assigned to the skin color decal texture, a bump map for the skin surface structure, a bump map for expressive wrinkles, and a *mask texture* that contains different monochrome masks such as a gloss map or a bump intensity map in its color channels. The bump map for the skin structure is computed directly from a synthetic human skin model similar to the one presented in [Ishii et al. 1993]. The wrinkles bump map is automatically created from the layout of the expressive wrinkles, taken from the skin texture. Hardware bump mapping for skin structure and wrinkles is implemented using the OpenGL `NV_vertex_program` and `NV_register_combiners` extensions. In addition, a gloss map is applied to specify the locally varying specular coefficient of a Blinn-Phong shading model, while the intensity of skin dimples over the face is controlled by a bump intensity map. Similar to a gloss map, the latter contains a scalar value per texel to specify the degree to which the bump mapped normal should affect the lighting computation. The whole process is carried out in a single rendering pass on the GeForce3, resulting in frame rates of about 100 fps. Figure 2 shows a comparison of a head model rendered with plain OpenGL capabilities and with our skin shading algorithm.

5 Landmark-based head deformation

Given the head model as described in the Section 3, we are using the landmark information to specify a deformation of the object space so that we can warp the complete head structure to a prescribed target landmark configuration. The details of this deformation are described in this section. In Sections 6 and 7 we demonstrate how this general method can be used for both creation and modification of an animatable head model.

5.1 Setting up the warp function

For deformation of biological tissues, BOOKSTEIN advocates an approach based on thin-plate splines, which minimizes the bending energy of a deformed surface [Bookstein 1997a]. The mechanism can be easily translated to the three-dimensional setting [Bookstein 1997b]. The theory is covered extensively in the literature, so we restrict to the practical construction of the deformation function.

The problem can be stated as one of interpolation: let $\mathbf{p}_i \in \mathbb{R}^3$ and $\mathbf{q}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$, be two sets of n landmarks. The *source landmarks* $\mathbf{p}_i$ lie on the geometry we want to deform, and the *target landmarks* $\mathbf{q}_i$ correspond to the features on the target head. We need to find a function $\mathbf{f}$ that maps the $\mathbf{p}_i$ to the $\mathbf{q}_i$:

$$\mathbf{q}_i = \mathbf{f}(\mathbf{p}_i), \quad i = 1, \ldots, n,$$

and which is defined on the volume spanned by the landmarks, so that the function can be used to deform all elements of the head structure. Such a mapping can be expressed by a radial basis function, i.e. a weighted linear combination of n basic functions ϕ_i defined by the source landmark points and an additional explicit affine transformation:

$$\mathbf{f}(\mathbf{p}) \;=\; \sum_{i=1}^{n} \mathbf{c}_i \phi_i(\mathbf{p}) \;+\; \mathbf{R}\mathbf{p} \;+\; \mathbf{t}, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ is a point in the volume, $\mathbf{c}_i \in \mathbb{R}^3$ are (unknown) weights, $\mathbf{R} \in \mathbb{R}^{3\times3}$ adds rotation, skew, and scaling, and $\mathbf{t} \in \mathbb{R}^3$ is a translation component. Following BOOKSTEIN, we simply use the biharmonic basic function $\phi_i(\mathbf{p}) := \|\mathbf{p} - \mathbf{p}_i\|_2$, which minimizes bending energy for the deformation [Duchon 1977]. We have the additional constraints

$$\sum_{i=1}^{n} \mathbf{c}_i = \mathbf{0} \quad \text{and} \quad \sum_{i=1}^{n} \mathbf{c}_i^T \mathbf{p}_i = 0$$

to remove affine contributions from the weighted sum of the basic functions [Pighin et al. 1998; Carr et al. 2001].

Setting up a system of linear equations relating source and target landmarks, the unknowns $\mathbf{R}$, $\mathbf{t}$, and $\mathbf{c}_i$ can be solved for simultaneously. We first construct three matrices:

$$\mathbf{B} = \begin{pmatrix} \mathbf{q}_1 & \ldots & \mathbf{q}_n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}^T \in \mathbb{R}^{(n+4)\times3},$$

$$\mathbf{P} = \begin{pmatrix} \phi_1(\mathbf{p}_1) & \ldots & \phi_n(\mathbf{p}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{p}_n) & \ldots & \phi_n(\mathbf{p}_n) \end{pmatrix} \in \mathbb{R}^{n\times n},$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{p}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{p}_n^T & 1 \end{pmatrix} \in \mathbb{R}^{n\times4}.$$

Now we set up a linear equation system of the form $\mathbf{AX} = \mathbf{B}$ with

$$\mathbf{A} = \begin{pmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{Q}^T & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+4)\times(n+4)},$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{c}_1 & \ldots & \mathbf{c}_n & \mathbf{R} & \mathbf{t} \end{pmatrix}^T \in \mathbb{R}^{(n+4)\times3}.$$

This linear system is solved using a standard LU decomposition with pivoting. We can now transform a point $\mathbf{p} \in \mathbb{R}^3$ according to Eq. (1).

5.2 Deforming the head structure

Given a warp function defined by landmarks placed on the skin of the source and target heads, we apply this function in different ways to the individual components of the model.

1. The *skin mesh* is deformed by direct application of the function to the vertices of the mesh.

2. The landmarks on the *skull mesh* are related to their counterparts on the skin by a vector, giving an offset from each skull landmark to the corresponding skin landmark, cf. Figure 1 d). When the skin geometry has been fixed, adjusting the local skin thickness thus amounts to changing the scale for such a vector, and deforming the skull mesh accordingly. The deformation function is obtained by offsetting the target skin landmarks along the negated new vectors, resulting in the desired new skull landmark positions. The warp from the current skull landmark positions to these new positions is then applied to the vertices of the skull mesh.

3. *Muscles* in our system are specified by a grid which is initially "painted" onto the skin. The actual shape of the muscles is computed automatically from the available space underneath the skin and follows the geometry of the skin surface. To transfer the muscles to the new geometry, we apply the deformation function to the grid vertices and re-compute the shape. The rebuild process also allows us to accommodate for changes in skin thickness.

4. For the other facial components eyes, teeth, and tongue, we only update position and scale automatically, due to their representation as rigid pieces of geometry in our system. Some fine-tuning is thus necessary to fit them exactly into the deformed model. In principle, if the components are also represented as meshes, the deformation can be applied to their vertices, making this manual step unnecessary.

6 Creating head models from range scans

A primary task in facial modeling is the reproduction of the heads of real individuals. One way to acquire the geometry of a head is to use a range scanning device. In practice, though, it turns out that there are a number of obstacles to using this geometry directly for an animatable model:

- the range data is often noisy and incomplete, especially for structured light scanners, due to projector/camera shadowing effects or bad reflective properties of the surface.

- the geometry is heavily oversampled: direct conversion to a triangle mesh regularly yields hundreds of thousands of polygons. For real-time animation, we need to reduce the complexity to about 3k polygons. Available mesh simplification techniques [Cignoni et al. 1998] unfortunately don't give enough control over the mesh connectivity to guarantee satisfyingly animatable models. Edges should be properly aligned to facial features and the mesh structure should reflect the basic symmetry of the face.

- some parts relevant for animation cannot be scanned, such as the inner part of the lips.

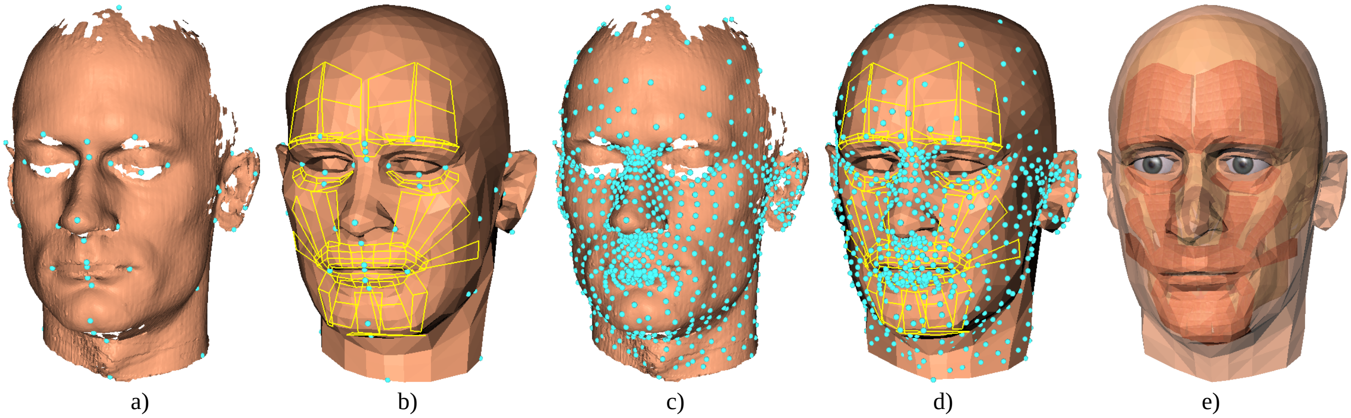


Figure 3: Adaptation of the reference mesh to scan data using feature mesh refinement: a) initial defective target mesh from range scans with landmarks added; b) source mesh and muscle outlines after first deformation; c) target geometry after three refinements; d) deformed source mesh after three refinements and warps; e) final mesh geometry with adapted skull and muscles.

For these reasons, a common solution is the use of a generic face mesh, which is fitted to the target geometry, cf. Section 2. We employ our landmark-based deformation approach to create an animatable model from the reference geometry. No parameterization of the scan data or the reference head mesh is needed, thus we are not restricted to scans from cylindrical range scanners [Lee et al. 1995], but can directly process data from arbitrary sources, and, concerning this, have no restrictions on the topology of the meshes. As a side effect of the procedure, the resulting target model is also tagged with a complete set of anthropometric landmarks, which can be directly used for further deformations, as will be demonstrated in Section 7.

The scan data is given in the form of a dense triangle mesh with no further fixing, such as hole filling, applied. We call this mesh the *target geometry* $\mathcal{M}^*$ in the following, while the reference head (including the structural components) is referred to as the *source geometry* $\mathcal{M}$ — the source geometry needs to be deformed to conform to the target. We proceed as follows:

1. $\mathcal{M}^*$ is tagged with a set of landmarks $\mathcal{L}^*$ corresponding to the set $\mathcal{L}$ defined on $\mathcal{M}$. This is a "computer-aided" interactive procedure.

2. a deformation is computed based on the landmark correspondences, and $\mathcal{M}$ is warped accordingly.

3. $\mathcal{L}$ and $\mathcal{L}^*$ are automatically refined to generate more correspondences.

4. The components of the reference head model, i.e. skull and muscles, are deformed to match $\mathcal{M}^*$ using the same deformation.

5. Repeat from Step 2 until convergence.

6. muscle shapes are rebuilt and the skull is warped once more to finally adjust the skull/skin relationship, as described in Section 5.2.

In Step 4 of the refinement loop, the skull is deformed using the skin mesh deformation to keep the relation between skin and skull within the accuracy determined by the current density of the landmark sets $\mathcal{L}$ and $\mathcal{L}^*$. Only in Step 6, the new skull/skin distance is asserted at the sparse locations where landmarks on skin and skull are paired.

We discuss the specification of the landmarks $\mathcal{L}^*$ and the landmark set refinement procedures in detail in the next two sections.

6.1 Specifying landmarks

Our method only requires the specification of a sparse set of landmarks on the target geometry $\mathcal{M}^*$. Due to the automatic refinement, face features do not need to be laboriously traced using dense correspondences or feature lines [Pighin et al. 1998]. The landmarks are taken from a standard set of the anthropometric literature and are thus well-defined and easy to identify, see Section 3. To ease the "repeated point-and-click" task, we again make use of the deformation function: given three or more landmarks specified manually, we can already set up a mapping from the source set of landmarks $\mathcal{L}$ to the target set $\mathcal{L}^*$. We then copy and warp all landmarks from $\mathcal{L}$ using this function, resulting in a rough approximation of the desired landmark distribution. Through manual inspection and correction more landmarks are repositioned and fixed in their target positions. The process can be iterated until all landmarks in $\mathcal{L}^*$ have assumed their intended positions. This simple method has shown to be particularly helpful in cases where the scan data is lacking a lot of shape information, since the copied landmarks will already be positioned in a meaningful way. Specifying our reference set of 60 landmarks thus takes only 10–20 minutes in practice. Figure 3 a) shows the scan tagged with the complete set of landmarks, cf. also Figure 1. If no further deformation using this specific set of landmarks is desired, the adaptation of the reference head to the scan data can be performed on an arbitrary set of landmarks.

To further automate the task, we are also actively experimenting with automatic mesh fitting methods, but we feel that there is currently no reliable way to automatically detect the features on the geometry with the required degree of accuracy, especially given the type of incomplete and noisy scan data we have: landmarks often must be placed manually in "mid-air" because the local geometry was not captured by the scanner. Simple heuristics [Noh and Neumann 2001] rely on well-behaved mesh geometry, and are not suitable for most of the anthropometric standard landmarks.

6.2 Adapting the generic mesh

After the initial deformation based on the user-specified landmarks, $\mathcal{M}$ and $\mathcal{M}^*$ are already in good correspondence, see Figure 3 a) and b). But, since the landmark distribution is very sparse, the details in the facial geometry of $\mathcal{M}^*$ are usually not well captured. We do not want to burden the user with specification of hundreds of feature points, so we have developed an automatic procedure that refines the landmark sets $\mathcal{L}$ and $\mathcal{L}^*$ and achieves as good a match as

```
$\mathcal{M}$ ← reference head mesh
$\mathcal{M}^*$ ← target head mesh
$\mathcal{L}$ ← landmark set on reference head
$\mathcal{L}^*$ ← landmark set on target head

$\mathcal{M}$ ← warp($\mathcal{L}, \mathcal{L}^*, \mathcal{M}$)        // deformation from $\mathcal{L}$ to $\mathcal{L}^*$ applied to $\mathcal{M}$

$\mathcal{F}$ ← feature_mesh($\mathcal{L}$)             // construct feature meshes
$\mathcal{F}^*$ ← feature_mesh($\mathcal{L}^*$)           // using landmark positions

repeat
  $(\mathcal{F}^*, B)$ ← subdivide($\mathcal{F}^*$)       // subdivide $\mathcal{F}^*$, store in $B$ baryc. coords
                                        // of new vertices w.r.t. parent triangles
  $(\mathcal{F}^*, D)$ ← project($\mathcal{F}^*, \mathcal{M}^*$) // project feature vertices onto surface of
                                        // $\mathcal{M}^*$ and store displacements in $D$

  $\mathcal{L}^*$ ← add_landmarks($\mathcal{F}^*, \mathcal{L}^*$) // more target landmarks for
                                        // appropriate new vertices in $\mathcal{F}^*$

  $\mathcal{F}$ ← subdiv_copy($\mathcal{F}, B, D$)     // subdivide $\mathcal{F}$ using $B$ and $D$

  $(\mathcal{F}, \mathcal{L})$ ← project($\mathcal{F}, \mathcal{M}$)      // project feature vertices, landmarks onto $\mathcal{M}$

  flip_edges($\mathcal{F}$)                    // improve feature mesh smoothness
  flip_edges($\mathcal{F}^*$)

  $\mathcal{M}$ ← warp($\mathcal{L}, \mathcal{L}^*, \mathcal{M}$)          // warp using new landmarks
until convergence
```

Table 1: The refinement algorithm for landmark sets on source and target geometries; see Section 6.2 for detailed explanation.

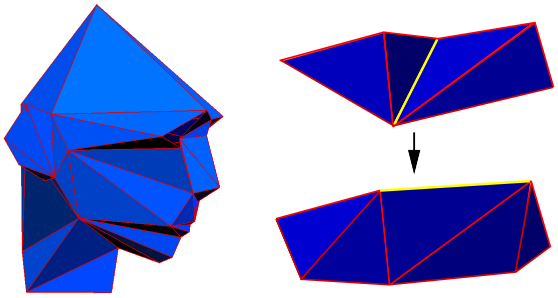


Figure 4: Left: a *feature mesh* is constructed by connecting the landmarks on the head geometry, forming a triangle mesh. Right: flipping edges after subdividing the mesh improves surface smoothness: new vertices on two neighboring subdivided triangles are connected by the yellow edge that previously divided the peaks.

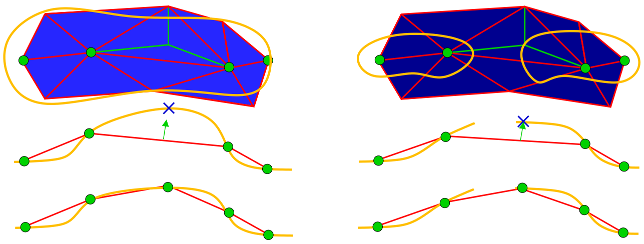


Figure 5: Refining corresponding triangles in the source (left, light blue) and target (right, dark blue) feature meshes. Top: the mesh is typically well-behaved in the source geometry, and fragmented in the target (orange curves). Where vertices of the target feature mesh project onto the geometry, landmarks have been added to both feature meshes (green dots). Target and source feature mesh triangles are refined equally (green edges). Middle: the normal displacement of the target mesh intersection is used to obtain a starting point for finding an intersection with the source mesh. Bottom: source geometry and feature meshes have been deformed to match the landmarks to the target.

possible in typically two or three iterations.[1] The algorithm outline is shown in Table 1. To be able to refine our standard set of landmarks automatically, we interpret the landmarks as the vertices of a triangle mesh, which we will call a *feature mesh* in the following. Figure 4 shows the layout of this mesh for the geometry of the reference head. One feature mesh $\mathcal{F}$ is constructed for $\mathcal{M}$, and another one $\mathcal{F}^*$ for $\mathcal{M}^*$, using the current landmark positions (so they are in fact identical after the first deformation). $\mathcal{F}^*$ is now refined by uniform subdivision: one vertex is inserted into each triangle, splitting it into three new triangles. This vertex is moved to the surface of $\mathcal{M}^*$, where we take care to find a correct sampling position on the surface, especially in areas of poor scan quality. Often, there is no part of the target surface in the vicinity of the new vertex. If there is, a new landmark is created at the vertex position, and added to $\mathcal{L}^*$. For each subdivided triangle, the refinement is encoded as the barycentric coordinate of the projection of the new vertex onto the parent triangle along the triangle normal. These coordinates are stored in a set B, and the corresponding scalar displacements along the normal vector in another set D. The right half of Figure 5 outlines the subdivision and projection step for one triangle of $\mathcal{M}^*$.

We now need to find counterparts on $\mathcal{M}$ for the newly created landmarks. Since $\mathcal{F}$ and $\mathcal{F}^*$ have the same structure, and the geometries are already in good alignment, we repeat the same refinement on $\mathcal{F}$, using the information from B and D. Each new vertex in $\mathcal{F}$ is now close to the source geometry, but usually not placed exactly on the surface, due to differences in facial detail. If there is a landmark for this vertex in $\mathcal{L}^*$, we find the nearest intersection along a ray starting at the vertex, correct its position and create a landmark at that point, adding it to $\mathcal{L}$. The left half of Figure 5 shows how this step is applied.

After all triangles have been refined in this manner, all edges in $\mathcal{F}$ and $\mathcal{F}^*$ from the previous generation are flipped, to improve the quality of the feature mesh surfaces, see Figure 4. The algorithm does not rely on the smoothness of the feature meshes, but if the change in surface normals between adjacent triangles can be kept small during refinement, this will improve the locality of the sampling of the surface. We also filter the triangle normals once by averaging with their neighbors.

Using the two refined versions of $\mathcal{L}$ and $\mathcal{L}^*$, we set up a new deformation function. Applying it to the source model, and to the corresponding feature mesh, results in a better approximation of the target geometry. The procedure is repeated, and the deformed mesh quickly stabilizes to a good fit of the target geometry. In practice, after only three iterations of the refinement procedure the geometry will have adapted optimally.

Since we use a triangle mesh generated directly from range scan data as the target geometry, we usually have to deal with large areas of the head geometry where there is no data, often interspersed with small specks of samples (e.g. the back of the head), see Figure 3 a). The refinement algorithm is thus geared towards finding these small “islands of data”, while being very conservative in accepting a sampling site on the geometry as a new landmark position. A wrongly placed landmark can cause large distortions in the deformed geometry, rendering it unusable, so we employ heuristics based on surface normals and landmark/surface distance to find and rank the acceptable sites, or reject creation of a landmark. The ray

[1] “As good as possible” here refers to the limits imposed by the discretization of the source mesh, which is usually much coarser than the scan data.

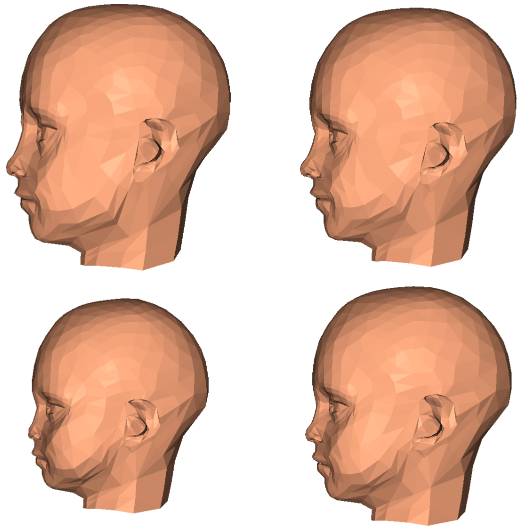


Figure 6: Geometric deformation of a boy's head by our constraint resolution technique. Clockwise from left top: 20 years, 12 years, 5 years (original age), 1 year.

intersection before repositioning a new vertex in the source feature mesh is much less critical, since we are operating on the deformed reference mesh, which has perfectly well-behaved geometry. Here, we just have to make sure not to intersect with backfacing parts of the geometry. Figure 3 shows the approximation of scanned head geometry by deformation of the reference head.

7 Growth and aging

A challenging problem in facial modeling and animation is the simulation of growth and aging. Besides the Arts, important applications exist in the forensic sciences and medicine: how does the child that got missing ten years ago look now? What are the long-term effects of a rhinoplastic operation? Often, a skilled artist is needed to draw conclusions towards age-related changes from photographs with the help of anthropometric data. Drastic changes in head size and facial proportions occur between childhood and maturity, as well as in skin texture and elasticity of the skin, not to mention hair growth. All of this affects the look of the face in its diverse static poses as well as in motion.

We demonstrate the application of growth and age transformations to the geometry of our animated head model using the deformation technique described in the previous sections. The set of landmarks on the model is used to obtain a variety of standard anthropometric measurements on the head, which are updated according to a user-specified change in age by a constraint-resolution mechanism. Our approach is inspired by DECARLO *et al.* [DeCarlo et al. 1998], but has no restrictions on the smoothness or parameterization of the surface. In fact, the computation of the age deformation uses only the landmark set, thus being independent from the deformed surface. In our system, we apply this deformation to the vertices of the head model's triangle mesh. Also, we do not operate on proportions, but only on distance measurements.

7.1 Landmark measurements

The landmarks on the head model correspond to those in the anthropometric literature used for head measurements. Specifically, we use tabulated measurements for a sample of the North American population of the Caucasian type, males and females between the ages of one year up to twenty-five years (after this age, there is almost no change in facial geometry due to growth) [Farkas 1994]. The data describes distance relations for pairs of landmarks, dependent on age and sex. Three types of distance measurements for a given landmark pair are used in the data:

- distances along one of the horizontal, vertical, and depth directions;
- Euclidean distance;
- arc length, traditionally measured using soft tape on the face.

The head model is placed in the standard posture used for anthropometric measurements, so the axis-aligned distances correspond to the x, y, and z axes in the local coordinate system. Each statistical measurement is given by its mean value μ and standard deviation σ. In our current system, we use 39 axis-aligned distance measurements, 25 Euclidean distances and 6 arc lengths, specified on a part of the standard landmark set on the reference head.

Given age and sex for the current head model, we first compute the current value d_c for a distance measurement directly from the landmarks on the model. The value is compared to the statistical mean value μ_c in the data tables to find its "position" in the assumed standard probability distribution. After looking up the mean value μ_t for the targeted age, we compute the final value d_t at the same relative position in the distribution:

$$d_t = \mu_t + \frac{\sigma_t}{\sigma_c}(d_c - \mu_c),$$

where σ_c and σ_t are standard deviations for the current and target age, respectively. Thus, we retain the characteristics of the source head model even over large changes in age.

For deforming the head geometry, we are now posed with the following problem: given the current landmark positions $\mathbf{p}_i$ and a number of distance measurements, what are the new landmark positions $\mathbf{q}_i$ $(i = 1, \ldots, n)$? This problem is largely underconstrained, as there are many solutions that fulfill these *hard constraints*. In our approach, we thus add more constraints between the landmarks that are closest to each other. After deformation, the distances between them should scale roughly with the global scaling s of the head, which we derive from the change in head height. These are *soft constraints*, in that they are used to find the best solution, but they are not strictly enforced.

7.2 Linear constraint resolution

Most of the distance measurements are given along one axis a $(a \in \{x, y, z\})$, which allows us to represent the problem as a set of linear constraints: we want to find the n new landmark coordinates $\mathbf{q}_{a,i}$, $i = 1, \ldots, n$, for each axis a. In the following, we derive a solution for one such axis.

The relation to the m hard distance constraints and $\widetilde{m}$ soft distance constraints can be expressed by a sparse linear system:

$$\mathbf{A}\mathbf{q}_a \;=\; \mathbf{d} + \sum_{i=1}^{\widetilde{m}} \lambda_i \widetilde{\mathbf{d}}_i \;,$$

where the combinatorial matrix $\mathbf{A} \in \mathbb{R}^{(m+\widetilde{m})\times n}$ specifies pairings of landmarks. Each row of $\mathbf{A}$ contains exactly two non-zero entries $+1$ and -1 in columns j and k, respectively, to denote the pairing of $\mathbf{q}_{a,j} - \mathbf{q}_{a,k}$. There can be at most $n(n-1)/2$ pairings of landmarks, but in practice, we have $m \approx 35$ plus $\widetilde{m} \approx 100$ landmark pairs per axis. The vector $\mathbf{d} \in \mathbb{R}^{m+\widetilde{m}}$ represents the hard constraints and has m non-zero entries in those positions, where the target distance is prescribed. Each vector $\widetilde{\mathbf{d}}_i \in \mathbb{R}^{m+\widetilde{m}}$ contains a single non-zero entry with the current distance $\mathbf{p}_{a,j} - \mathbf{p}_{a,k}$ between a pair of landmarks not constrained by $\mathbf{d}$ in the corresponding position, i.e. in the row where $\mathbf{A}$ specifies the pairing of $\mathbf{q}_{a,j} - \mathbf{q}_{a,k}$. Since we want to enforce the hard constraints $\mathbf{d}$ given by the data, but keep the soft distances between the other landmarks close to what they were, we solve for weights λ_i close to the global scaling factor s.

The system can be easily reformulated by shifting terms to include the $\widetilde{\mathbf{d}}_i$ and λ_i in the matrix:

$$\begin{pmatrix} \mathbf{A} & \widetilde{\mathbf{D}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{q}_a \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{s} \end{pmatrix}, \qquad (2)$$

where the columns of $\widetilde{\mathbf{D}} \in \mathbb{R}^{(m+\widetilde{m})\times\widetilde{m}}$ are composed of the vectors $-\widetilde{\mathbf{d}}_i$, and $\boldsymbol{\lambda}$ is a vector built from the λ_i in the same order. The submatrix $\mathbf{I} \in \mathbb{R}^{\widetilde{m}\times\widetilde{m}}$ is an identity matrix. On the right hand side, $\mathbf{s}$ has $\widetilde{m}$ entries with the constant scaling factor s. The system is now overconstrained and we solve for the $\mathbf{q}_{a,i}$ and λ_i using a singular value decomposition (SVD) [Press et al. 1992]. Clamping the singular values to achieve a prescribed condition number for the least squares problem before back-substitution allows us to get rid of linearly dependent constraints, as they occur in practice.

According to this method, we set up and solve three independent linear systems for the distance constraints along the x, y, and z axes. Since the data is only as exact as the procedures used for taking measurements of the sample population, and the collected data is only statistical, a precise solution for a given individual head can not in general be achieved. However, SVD will give a best fit in the least squares sense: for a system $\mathbf{Ax} = \mathbf{b}$, the solution vector $\mathbf{x}$ that minimizes the residual error $\|\mathbf{Ax} - \mathbf{b}\|_2$ will be found. In Eq. (2), the values in $\mathbf{d}$ are typically in the range of 10–200 mm, while the values in $\mathbf{s}$ are close to 1.0. Thus, a small error (i.e. a displacement from the ‘ideal’ target position) in one of the new landmark coordinates $\mathbf{q}_{a,i}$ results in a much larger residual error than a small deviation in one of the weights λ_i. As a result, we found the hard constraints to be fulfilled with a maximum absolute error of two millimeters in our experiments.

7.3 Non-linear constraints

For some landmark pairs, a Euclidean distance is given, which can not be included directly into the linear system. We linearize the problem by splitting such a constraint into three axis-aligned constraints. Given a current vector from one landmark to another and a prescribed target distance between them, we assume that the direction of that vector will not change drastically in the solution. We scale this vector to the target length and project it onto the three axes of the global coordinate system. We add the three projected distances as additional linear constraints into the equations described in the previous section and let SVD run as before, arriving at a solution that approximately fulfills the Euclidean distance constraint. To improve the solution, we repeat the projection process and solve again, until convergence. In practice, three iterations suffice.

Arc lengths are another measurement in the data. Since the pairs connected by arcs are constrained additionally by distance measurements in the table data, we do not include the arc lengths in the constraint resolution mechanism. Instead, we use the arc measurements only to improve the shape of the surface *after* solving. The arc is approximated by a circle segment: we use a virtual “middle landmark” that is placed between the two landmarks connected by the arc. This landmark is shifted along the surface normal, to give the circle segment the arc length specified in the data.

8 Results and conclusion

We have presented a head model with anatomical structure and deformation methods that allow adaptation of such a model to individual scan data, and simulation of human head growth. Obviously, without medical volumetric data, the finer individual details of the head anatomy cannot be guessed by our method. Still, without such expensive medical hardware as a CT scanner, we are able to produce a more plausible reconstruction of these internal structures than previous approaches that estimate the skull as an offset surface from the skin or use simpler models of muscle geometry.

The landmark-based fitting has shown to work robustly on raw meshes obtained from range scans of a variety of individuals. The interactive specification of the initial sparse landmark set has shown to be advantageous for creating models from incomplete data, where large parts of the geometry cannot be estimated automatically in a reliable way. The robust adaptation to this kind of input data makes mesh fixing and hole filling abdicable, a process that can otherwise easily take several hours of interactive work.

The computational cost of the landmark refinement algorithm largely depends on ray/mesh intersection tests and point/triangle distance computations. We expect a great speed-up from optimization of these tests. In our current implementation, we arrive at about five minutes total run time for the fitting process on a 1 GHz PC with a 100k triangle target mesh. Given the scan data, the whole process of creating an animatable head model including the tuning of eye, teeth, and tongue positions takes 20–30 minutes in our experience.

Further deformation of the face for changing the age produces plausible results, which is encouraging given that only a small amount of statistical data is used. Our method assumes that a measurement keeps its variance from the statistical mean over the years: a nose that is relatively big for an adult, is assumed to be big in childhood. Together with the scarcity of the facial measurements, this tends to retain the characteristics of the original face to a sometimes too strong degree. Also, examination of the tabulated arc measurements delivered surprising results: the ratio between arc length and corresponding Euclidean distance remains almost constant through the ages in the table data, i.e. the roundness of the face does not vary significantly, different from what one would expect especially for younger faces. To incorporate the “puffiness” of small childrens’ cheeks that can be observed in the real world, we allowed for an adjustable slight increase in the arc length up to 10 % over the original values for a young child.

Transfer of expressions and animations between generated models using the same initial muscle set has shown to work well even over widely varying age ranges. The common muscle set and parameterization of all our head models simplify the creation and use of a generic expression library a great deal. After adaptation of the model, this parameterization can be used for further editing, which is an advantage over purely geometric transfer of motion between models [Noh and Neumann 2001]: individual characteristics of facial motion, such as a particular way of smiling, can be included by manipulation of the same parameters, instead of working on the vertex data of the new model. Also, we see some automation potential for dealing with age-related characteristics based on the muscle parameterization: e.g. for a young child, we globally scale down facial muscle movements, to accommodate for the not yet fully developed abilities of expression.

9 Future work

Skin stiffness constants, skin thickness and muscle layer thickness are evaluated and used for each deformation of the reference head to re-shape the skull and the muscle layer, and to initialize the spring mesh. We currently have to adjust these parameters manually to accommodate for individual and age-related changes. Given more statistical data, it should be possible to have these values computed automatically. Also, the visual impact of such adjustments onto the actual animation needs to be examined.

For further automation and enhancement of the precision of the landmark specification process, we would like to make use of the information contained in the photographs used for texturing. It also is appealing to use the landmarks on the fitted face model for this texturing step, which is currently a separate procedure. This would also apply to the synthesis of wrinkles and skin structure. We are investigating if image-based techniques such as proposed by TIDDEMAN *et al.* [Tiddeman et al. 2001] can be used to minimize manual texture processing for wrinkle definition.

Currently, we cannot predict the skin color of an individual at an age different from the age of data acquisition. Some heuristics could be applied: babies of Caucasian type typically have a pale, pinkish skin color, while adults often have suntanned skin. Some more reliable results could probably be obtained by applying age-dependent skin parameters such as moisture or elasticity to a micro geometry skin model and using a BRDF-based rendering approach.

Our results indicate that "anthropometric modeling" is a fruitful approach, and could become a useful tool for artists and scientists alike. More detailed and precise data, and a better understanding of age-related changes contained in this data are needed, though. Based on our constraint resolution technique, new face modeling tools can be devised that allow specification of facial feature relations either directly, or indirectly by age, gender, or other statistically captured variables.

References

ALBRECHT, I., HABER, J., AND SEIDEL, H.-P. 2002. Speech Synchronization for Physics-based Facial Animation. In *Proc. WSCG 2002*, 9–16.

BLANZ, V., AND VETTER, T. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Computer Graphics (SIGGRAPH '99 Conf. Proc.)*, 187–194.

BOOKSTEIN, F. L. 1997. *Morphometric Tools for Landmark Data*. Cambridge University Press.

BOOKSTEIN, F. L. 1997. Shape and the Information in Medical Images: A Decade of the Morphometric Synthesis. *Computer Vision and Image Understanding 66*, 2, 97–118.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and Representation of 3D Objects With Radial Basis Functions. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, ACM SIGGRAPH, 67–76.

CHADWICK, J. E., HAUMANN, D. R., AND PARENT, R. E. 1989. Layered Construction for Deformable Animated Characters. In *Computer Graphics (SIGGRAPH '89 Conf. Proc.)*, vol. 23, 243–252.

CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1998. A comparison of mesh simplification algorithms. *Computers & Graphics 22*, 1, 37–54.

DECARLO, D., METAXAS, D., AND STONE, M. 1998. An Anthropometric Face Model using Variational Techniques. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, 67–74.

DUCHON, J. 1977. Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller, Eds., vol. 571 of *Lecture Notes in Mathematics*, 85–100.

FARKAS, L. G. 1994. *Anthropometry of the Head and Face*, 2nd ed. Raven Press.

HABER, J., KÄHLER, K., ALBRECHT, I., YAMAUCHI, H., AND SEIDEL, H.-P. 2001. Face to Face: From Real Humans to Realistic Facial Animation. In *Proc. Israel-Korea Binational Conference on Geometrical Modeling and Computer Graphics*, 73–82.

ISHII, T., YASUDA, T., YOKOI, S., AND TORIWAKI, J. 1993. A Generation Model for Human Skin Texture. In *Proc. Computer Graphics International '93*, 139–150.

KÄHLER, K., HABER, J., AND SEIDEL, H.-P. 2001. Geometry-based Muscle Modeling for Facial Animation. In *Proc. Graphics Interface 2001*, 37–46.

KOCH, R. M., GROSS, M. H., AND BOSSHARD, A. A. 1998. Emotion Editing using Finite Elements. In *Computer Graphics Forum (Proc. Eurographics '98)*, vol. 17, C295–C302.

LANITIS, A., TAYLOR, C., AND COOTES, T. 1999. Modeling the process of ageing in face images. In *Proc. 7th IEEE International Conference on Computer Vision*, IEEE, vol. I, 131–136.

LEE, Y., TERZOPOULOS, D., AND WATERS, K. 1993. Constructing Physics-based Facial Models of Individuals. In *Proc. Graphics Interface '93*, 1–8.

LEE, Y., TERZOPOULOS, D., AND WATERS, K. 1995. Realistic Modeling for Facial Animation. In *Computer Graphics (SIGGRAPH '95 Conf. Proc.)*, 55–62.

LEE, W.-S., WU, Y., AND MAGNENAT-THALMANN, N. 1999. Cloning and Aging in a VR Family. In *Proc. IEEE Virtual Reality '99*, 13–17.

NOH, J., AND NEUMANN, U. 2001. Expression cloning. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, ACM SIGGRAPH, 277–288.

PIGHIN, F., HECKER, J., LISCHINSKI, D., SZELISKI, R., AND SALESIN, D. H. 1998. Synthesizing Realistic Facial Expressions from Photographs. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, 75–84.

PLATT, S. M., AND BADLER, N. I. 1981. Animating Facial Expressions. In *Computer Graphics (SIGGRAPH '81 Conf. Proc.)*, vol. 15, 245–252.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2. ed. Cambridge University Press, Cambridge, MA.

SCHEEPERS, F., PARENT, R. E., CARLSON, W. E., AND MAY, S. F. 1997. Anatomy-Based Modeling of the Human Musculature. In *Computer Graphics (SIGGRAPH '97 Conf. Proc.)*, 163–172.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-Form Deformation of Solid Geometric Models. In *Computer Graphics (SIGGRAPH '86 Conf. Proc.)*, vol. 20, 151–160.

SZELISKI, R., AND LAVALLÉE, S. 1996. Matching 3-D Anatomical Surfaces with Non-Rigid Deformations using Octree-Splines. *International Journal of Computer Vision 18*, 2, 171–186.

TERZOPOULOS, D., AND WATERS, K. 1990. Physically-based Facial Modelling, Analysis, and Animation. *Journal of Visualization and Computer Animation 1*, 2 (Dec.), 73–80.

TIDDEMAN, B., BURT, M., AND PERRETT, D. 2001. Prototyping and Transforming Facial Textures for Perception Research. *IEEE Computer Graphics and Applications 21*, 5 (Sept./Oct.), 42–50.

TURNER, R., AND GOBBETTI, E. 1998. Interactive construction and animation of layered elastically deformable characters. *Computer Graphics Forum 17*, 2, 135–152.

VESELY, F. J. 1994. *Computational Physics: An Introduction*. Plenum Press, New York.

WATERS, K., AND FRISBIE, J. 1995. A Coordinated Muscle Model for Speech Animation. In *Proc. Graphics Interface '95*, 163–170.

WATERS, K. 1987. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics (SIGGRAPH '87 Conf. Proc.)*, vol. 21, 17–24.

WILHELMS, J., AND VAN GELDER, A. 1997. Anatomically Based Modeling. In *Computer Graphics (SIGGRAPH '97 Conf. Proc.)*, 173–180.

WU, Y., MAGNENAT-THALMANN, N., AND THALMANN, D. 1994. A Plastic-Visco-Elastic Model for Wrinkles in Facial Animation and Skin Aging. In *Proc. Pacific Graphics '94*, 201–214.

WU, Y., KALRA, P., MOCCOZET, L., AND MAGNENAT-THALMANN, N. 1999. Simulating Wrinkles and Skin Aging. *The Visual Computer 15*, 4, 183–198.

Reanimating the Dead: Reconstruction of Expressive Faces from Skull Data

Kolja Kähler[*] Jörg Haber[†] Hans-Peter Seidel[‡]

MPI Informatik, Saarbrücken, Germany

Abstract

Facial reconstruction for postmortem identification of humans from their skeletal remains is a challenging and fascinating part of forensic art. The former look of a face can be approximated by predicting and modeling the layers of tissue on the skull. This work is as of today carried out solely by physical sculpting with clay, where experienced artists invest up to hundreds of hours to craft a reconstructed face model. Remarkably, one of the most popular tissue reconstruction methods bears many resemblances with surface fitting techniques used in computer graphics, thus suggesting the possibility of a transfer of the manual approach to the computer. In this paper, we present a facial reconstruction approach that fits an anatomy-based virtual head model, incorporating skin and muscles, to a scanned skull using statistical data on skull / tissue relationships. The approach has many advantages over the traditional process: a reconstruction can be completed in about an hour from acquired skull data; also, variations such as a slender or a more obese build of the modeled individual are easily created. Last not least, by matching not only skin geometry but also virtual muscle layers, an animatable head model is generated that can be used to form facial expressions beyond the neutral face typically used in physical reconstructions.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation G.3 [Probability and Statistics]—Multivariate statistics G.1.2 [Numerical Analysis]: Approximation—Approximation of surfaces and contours

Keywords: facial modeling, forensic art, face reconstruction

1 Introduction

1.1 Background

For well over a hundred years, forensic art and science has been assisting law enforcement. One of the major areas of concern in this area is facial reconstruction for postmortem identification of humans from their physical remains. Manual reconstruction and identification techniques build on the tight shape relationships between the human skull and skin: for instance, the presumed identity of a murder victim can be confirmed by superimposing a facial photograph with a properly aligned and sized image of the skull. If no photograph is available, the look of the face can be reconstructed to a certain degree by modeling the missing tissue layers directly onto the skull or a plaster cast made from it.

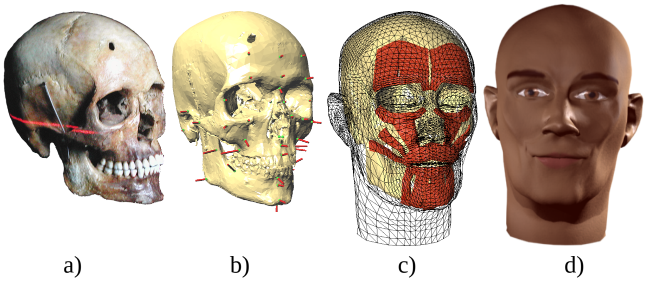


Figure 1: Reconstruction of a face from the skull: a) scanning the skull; b) skull mesh tagged with landmarks; c) skin mesh with muscles fitted to the skull; d) textured skin mesh, smiling expression.

The first documented case using three-dimensional facial reconstruction from the skull dates back to 1935 [Taylor 2001]. A key experiment was later performed by KROGMAN [1946]: given the body of a deceased person, he took a picture of the cadaver head before extracting the skull. The skull was provided to a sculptor along with information about sex, origin, and age of the late owner, plus data on the average tissue thicknesses at several positions in the face. From this material, a reconstruction sculpture was created that could be compared to the original head. Since that time, three-dimensional facial reconstruction from the skull has been much refined, but the method has essentially remained the same. Researchers have examined the skull / skin relationships for different ethnic groups [Lebedinskaya et al. 1993] and analyzed the correspondences of skull morphology and facial features [Fedosyutkin and Nainys 1993]. Others found correlations between muscle activity and skull shape [Moore and Lavelle 1974; Weijs and Hillen 1986]. In her comprehensive textbook, TAYLOR [2001] describes the craft in great detail.

Much of the fascination of the topic is due to the combined efforts of science and art, resulting in often astonishingly lifelike reconstructions, given the little available input (see Fig. 2). Many parameters of the outward appearance of an individual cannot be readily derived from the skull, though. The process is thus highly dependent on rules of thumb, the experience of the artist, and some guesswork. It is, for instance, next to impossible to reconstruct the shape of the ears based on scientific reasoning, although empirically there seems to be a relation of ear height to the length of the nose.

1.2 The Manual Reconstruction Process

The traditional work process for facial reconstruction begins with preparation of the skull. Since the skull is often evidence in a criminal case, great care needs to be taken in handling it: some parts are extremely thin and fragile, especially in the nose and the orbits. For identification, the teeth often provide a lot of useful informa-

[*] e-mail: kkaehler@acm.org
[†] e-mail: haberj@acm.org
[‡] e-mail: hpseidel@mpi-sb.mpg.de

appeared originally in ACM Transactions on Graphics (Proc. SIGGRAPH 2003), 22(3), 2003

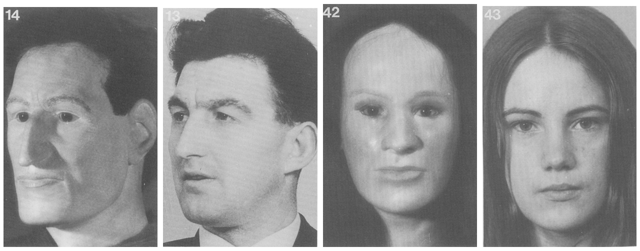


Figure 2: Comparison of sculpted reconstructions with photographs. Left: male subject; right: female subject. (Images: Copyright ©[Helmer et al. 1993], reprinted by permission of Wiley-Liss, Inc., a subsidiary of John Wiley & Sons, Inc.)

tion, so a dental analysis is usually performed at this stage. For the reconstruction of the lower face, the mandible needs to be properly aligned and secured to the skull. In cooperation with an anthropologist, and possibly given more information from the remains of the victim, an estimation of age, ancestry, sex, and stature can now be obtained.

The actual face reconstruction proceeds with one of two available approaches: the *anatomical method* and the *tissue depth method*. The anatomical method attempts reconstruction by sculpting muscles, glands, and cartilage, fleshing out the skull layer by layer. This technique is more often used in the reconstruction of fossil faces, where no statistical population data exists [Zollikofer et al. 1998]. As TAYLOR states, this technique is very time consuming, occupying "many hundreds of hours". It also requires a great deal of detailed anatomical knowledge. Therefore, the alternative tissue depth method has become the more popular reconstruction technique in law enforcement. Here, standard sets of statistical tissue thickness measurements at specific points on the face are used. Each measurement describes the total distance from skin surface to the skull, including fat and muscle layers. The method is thus more rapid than the anatomical method and does not require as much anatomical knowledge. Such measurements have been collected for males and females of several racial groups, using needles, X-rays, or ultrasound techniques. The tissue depth data most often used by police artists today was collected primarily by RHINE *et al.* [Rhine and Campbell 1980; Rhine and Moore 1984]. The data is sorted into "slender", "normal", and "obese" groups, as well as by sex and race.

Given the set of measurements, tissue depth markers are now placed on the skull or a cast made from it, reflecting the tissue thickness at the sample points. These markers are oriented orthogonally to the skull surface, corresponding to the direction of the tissue thickness measurements. Using the markers and other features on the skull for guidance, the face is modeled on top of the skull using clay. A snapshot of the beginning stages of a reconstruction using the tissue depth method is shown in Fig. 3.

1.3 Our approach

Looking at the facial reconstruction process as described above from a computer graphics perspective, it essentially boils down to a surface interpolation problem. We thus implement the manual "dowel placement" method as an interactive procedure, obtaining position and distance constraints that define the relation between skin and skull at selected sample positions. The sculpting of the skin surface is mapped to a volume deformation applied to a head model template, satisfying these constraints. The deformation approach has the additional advantage of being applicable to additional structures attached to the template: in our system, we map a muscle structure to the fitted head model (see Fig. 1), enabling animation on the reconstructed head in a physics-based facial animation framework.

The remainder of this paper is organized as follows: after reviewing related work in Section 2, we discuss acquisition of skull data and interactive landmark placement for setting up surface constraints in Section 3. Section 4 describes the structure of our generic head model and how it is fitted to the skull. Animation and texture generation for the resulting head model are touched upon in Section 5. We present examples in Section 6 and draw conclusions from our results in Section 7.

2 Previous and Related Work

2.1 Computer-Aided Face Reconstruction

Perhaps due to the lack of rigid taxonomies and hard rules, the use of computers and computer graphics in this forensic application is still very limited. The procedures described above cannot be cast easily into a computer program that produces good results in an automated manner—the experience and judgment of the practitioner remain a vital part of the system.

In law enforcement practice, computer-aided techniques restrict to relatively simple image and video manipulation: face photographs are used for skull superimposition [Grüner 1993; Miyasaka et al. 1995], while image warping and retouching enable a basic simulation of aging [Taylor 2001, p. 253]. This situation is unfortunate, since the traditional three-dimensional face reconstruction process is extremely time-consuming and expensive. It is hardly feasible to produce a variety of different plausible reconstructions from one skull, simply due to the effort that has to be put into the creation of each model. Also, repeated physical handling of the original skull increases the risk of damage.

One prototypical computer-based face reconstruction system, allowing fitting of a generic hierarchical B-spline head model to a skull mesh, is described by ARCHER in her Master's thesis [1997]. The user places dowels on a skull model with prescribed tissue thickness values, resulting in targets for a B-spline surface fitting process. The interpolation process is tricky and requires careful preparation of the template head model.

In the approach presented by MICHAEL and CHEN [1996], a source head model H_s that includes a skull S_s is deformed using a volume distortion function V such that the deformed source skull approximately matches the target skull S_t: $V(S_s) \approx S_t$. It is assumed that the deformed source head model $V(H_s)$ bears a good resemblance to the (unknown) target head model. The volume distortion function V is set up as a field warp using fourty pairs of disc fields, which are manually placed around the skull. No details are given about the placement of these control fields.

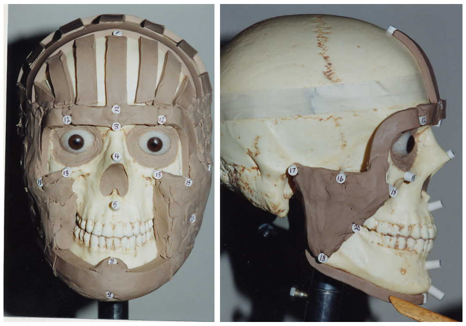


Figure 3: Modeling the face with clay on top of the skull using the tissue depth method. (Images [Taylor 2001], reprinted by permission.)

A deformation technique similar to the one used in our approach is employed by VANEZIS *et al.* [2000]. A facial template chosen from a database of scanned faces is deformed to match the position of target face landmarks, which have been derived from adding statistical tissue thickness values to the corresponding skull landmarks. The resulting reconstructed heads are not always complete (for instance, the top of the head is usually missing). The authors suggest to export an image of the reconstructed head and to apply a final image-processing step to add eyes, facial and head hair.

The above methods require a lot of manual assistance in setting up the interpolation function [Archer 1997; Michael and Chen 1996], or rely on a database of head templates [Vanezis et al. 2000]. In contrast, we develop reconstructions from one head template with relatively few markers, and use additional mechanisms to improve reconstruction results (see Section 4.3). Our approach always generates complete head models. Instead of using higher-order surfaces or point samples, the surface of our deformable head template is an arbitrary triangle mesh, simplifying later artistic modifications of the result using standard modeling tools. To the best of our knowledge, integration of expressive facial animation is not discussed by any other computer-aided facial reconstruction approach.

Other than explicit treatment of facial reconstruction, the creation of virtual head models based on human anatomy is well researched and documented in the computer graphics literature. Major developments in this area are discussed in the following section.

2.2 Human Head Modeling

A variety of techniques exists to create a face model from images or scan data. In the method presented by LEE *et al.* [1995], animatable head models are constructed semi-automatically from range scans. A generic face mesh with embedded muscle vectors is adapted to range scans of human heads. This process relies on the planar parameterization of the range scans as delivered, for instance, by the Cyberware digitizers. PIGHIN *et al.* [1998] interactively mark corresponding facial features in several photographs of an individual to deform a generic head model using radial basis functions. Animation is possible by capturing facial expressions in the process and blending between them. CARR *et al.* [2001] use radial basis functions to generate consistent meshes from incomplete scan data. Employing a large database of several hundred scanned faces, BLANZ *et al.* [1999] are able to create a geometric head model from only a single photograph. This model has the same resolution as the range scans in the database and cannot be readily animated. In the context of medical imaging, SZELISKI *et al.* [1996] minimize the distance between two surfaces obtained from volume scans of human heads by applying local free-form deformations [Sederberg and Parry 1986] and global polynomial deformations. The method does not require specification of corresponding features on the geometries.

Several facial animation systems use an approximation of the layered anatomical structure. WATERS [1987] represents skin and muscles as separate entities, where muscle vectors and radial functions derived from linear and sphincter muscles specify deformations on a skin mesh. In contrast to this purely geometric technique, physics-based approaches attempt to model the influence of muscle contraction onto the skin surface by approximating the biomechanical properties of skin. Typically, mass-spring or finite element networks are used for numerical simulation [Platt and Badler 1981; Lee et al. 1995; Koch et al. 1998]. From an initial triangle mesh, TERZOPOULOS and WATERS [1990] automatically construct a layered model of the human face. The model structure consists of three layers representing the muscle layer, dermis, and epidermis. The skull is approximated as an offset surface from the skin. Free-form deformations are employed by CHADWICK *et al.* [1989] to shape the skin in a multi-layer model, which contains bones, muscles, fat tissue, and skin. SCHEEPERS *et al.* [1997] as well as WILHELMS and VAN GELDER [1997] introduce anatomy-based muscle models for animating humans and animals, focusing on the skeletal musculature. Skin tissue is represented only by an implicit surface with zero thickness [Wilhelms and Van Gelder 1997].

We build our system on the deformable, anatomy-based head model described by KÄHLER *et al.* [2002]. There, a generic face mesh with underlying muscle and bone layers is deformed to match scanned skin geometry. This process is adopted here to match the muscle and skin layers to given skull data instead.

3 Preparation of the Skull

Our approach uses three-dimensional skull data acquired, for instance, from volume scans and extraction of the bone layers, or by range scanning a physical skull. The test data used for the examples in Section 6 was acquired using both types of scans. To speed up processing, a triangle mesh of the skull model comprised of 50-250k polygons is produced by mesh decimation techniques [Garland and Heckbert 1997]. In general, the original data should be simplified as little as possible since minute details on the skull can give important clues for the reconstruction. The mesh resolution is chosen for adequate responsiveness of our interactive skull editor application. In practice, it is helpful to have the original data set (or the physical skull) ready as a reference during editing.

In the editor, the skull model is equipped with landmarks, as shown in Fig. 4. Points on the skull surface are simply picked to create a landmark, which can then be moved around on the surface for fine positioning. Each landmark is associated with a vector in surface normal direction, corresponding to the typical direction of thickness measurements. As can be seen on the right image in Fig. 4, some skull / skin correspondences are in fact non-orthogonal to the skull surface in the area of the lips. This is corrected for at a later step of the fitting process, as described in Section 4.3. The landmark vector is scaled to the local tissue thickness, which is looked up automatically by the landmark's assigned name in a table based on RHINE's data (see Section 1.2). The specific set of landmarks used in our system is listed in Appendix A.

4 Fitting the Deformable Head Model

4.1 Head Model Structure

When the skull is tagged with landmarks, it serves as the target for deformation of the generic head model shown in Fig. 5. Since the head model is used in a physics-based animation system, it does

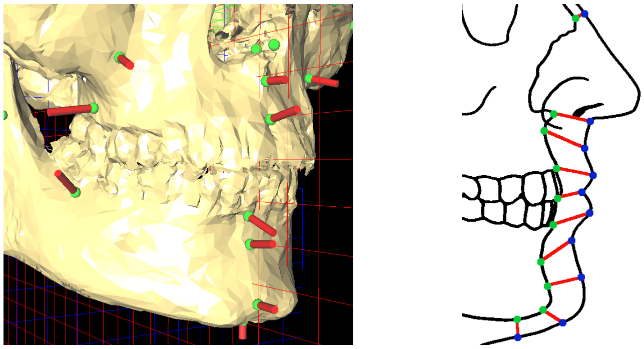


Figure 4: Skull landmark specification in the mouth area. Left: snapshot from our landmark editor; right: correspondences between skull and skin markers (Image after [y'Edynak and İşcan 1993])

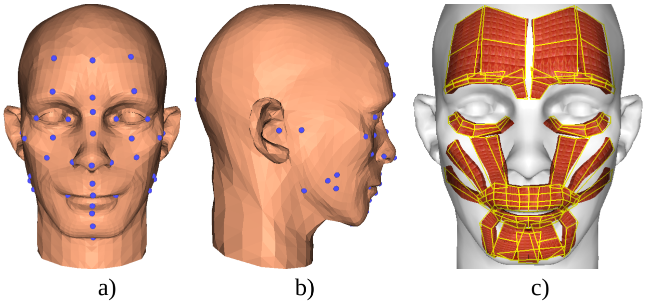


Figure 5: The deformable head model: a) head geometry with landmarks (blue dots), front view; b) side view; c) underlying muscles (red) created from layout grids (yellow).

not only consist of the visible outer geometry. The encapsulated structure includes:

the skin surface represented as a triangle mesh. The mesh resolution should be high enough to ensure good fitting results. Our template head mesh consists of 8164 triangles.

virtual muscles to control the animation. Each muscle is specified by a grid laid out on the skin, the actual muscle shape being computed automatically to fit underneath the skin surface. Each muscle consists of an array of fibers, which can contract in a linear or circular fashion. Our model includes 24 facial muscles responsible for facial expressions. Fig. 5(c) shows the muscle layout on the head template.

a mass-spring system connecting skin, muscles, and skull, built after the head model is fitted to the skull. For animation, muscles pull at spring nodes attached to their surface, in turn causing deformation of the spring mesh in the skin surface layer.

landmarks defined on the skin surface, as shown in Fig. 5(a) and (b). The majority of these landmarks corresponds to the landmarks interactively specified on the skull. These landmark pairs control the basic fitting of the head structure as described in Section 4.2. A few additional landmarks are only defined on the skin and are used for the final adjustments of the reconstructed shapes discussed in Section 4.3.

The head model is similar to the one in [Kähler et al. 2002], where detailed descriptions of the muscle model and animation approach can also be found.

4.2 Landmark-Based RBF Deformation

Given the deformable head model with n predefined skin landmark positions $\mathbf{p}_i \in \mathbb{R}^3$ and the corresponding landmarks $\mathbf{s}_i \in \mathbb{R}^3$ ($i = 1,\dots,n$) specified on the skull, we set up a space deformation that fits the skin and the muscle layout to the skull.

The target skull landmarks have associated tissue depth vectors $\mathbf{d}_i$, so corresponding skin landmark positions $\mathbf{q}_i$ are defined as

$$\mathbf{q}_i = \mathbf{s}_i + \mathbf{d}_i.$$

The problem can now be treated as one of interpolation: we need to find a function $\mathbf{f}$ that maps the $\mathbf{p}_i$ to the $\mathbf{q}_i$:

$$\mathbf{q}_i = \mathbf{f}(\mathbf{p}_i), \quad i = 1,\dots,n.$$

The unknown function $\mathbf{f}$ can be expressed by a radial basis function, i.e., a weighted linear combination of n basic functions ϕ_i and an additional explicit affine transformation:

$$\mathbf{f}(\mathbf{p}) \;=\; \sum_{i=1}^{n} \mathbf{c}_i \phi_i(\mathbf{p}) + \mathbf{R}\mathbf{p} + \mathbf{t}, \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ is a point in the volume, $\mathbf{c}_i \in \mathbb{R}^3$ are (unknown) weights, $\mathbf{R} \in \mathbb{R}^{3\times3}$ adds rotation, skew, and scaling, and $\mathbf{t} \in \mathbb{R}^3$ is a translation component. The ϕ_i are defined by the source skin landmark points. According to BOOKSTEIN [1997], for deformation of biological solids an approach based on thin-plate splines is favorable. We thus use the simple biharmonic basic function $\phi_i(\mathbf{p}) := \left\|\mathbf{p} - \mathbf{p}_i\right\|_2$, which minimizes bending energy for the deformation [Duchon 1977].

To remove affine contributions from the weighted sum of the basic functions [Pighin et al. 1998; Carr et al. 2001], we include the additional constraints

$$\sum_{i=1}^{n} \mathbf{c}_i = \mathbf{0} \quad \text{and} \quad \sum_{i=1}^{n} \mathbf{c}_i^T \mathbf{p}_i = 0.$$

The resulting system of linear equations is solved for the unknowns $\mathbf{R}, \mathbf{t}$, and $\mathbf{c}_i$ using a standard LU decomposition with pivoting, to obtain the final warp function $\mathbf{f}$. This function can now be used according to Eq. (1) to transform a point $\mathbf{p}$ in the volume spanned by the landmarks. We apply $\mathbf{f}$ to the skin and muscle components of the generic model in the following ways:

- The *skin mesh* is deformed by direct application of the function to the vertices of the mesh.
- The *muscles* are transferred to the new geometry by warping their layout grid vertices, followed by recomputation of the shape to fit the deformed skin mesh.

Since our landmark set is comprised of only 40 landmarks (see Appendix A), the computed deformation doesn't properly align the skin to the skull in all places, as can be seen in Fig. 6(a). Interactive specification of more landmarks puts an undesirable additional burden onto the user, so additional landmark pairs are computed automatically by interpolation between existing ones on the upper and back part of the cranium, as well as on the mandible, as shown in Fig. 6(b). The thickness value of an interpolated skull landmark is also interpolated, where only such skull areas are chosen for landmark interpolation where the tissue thickness is near-constant. Tissue depth interpolation would be problematic, for instance, in the mid-face area, where thickness values change drastically from the cheekbone to the mid-face region below.

4.3 Additional Reconstruction Hints

The tissue depth values at the marker positions define the basic shape of the reconstructed head, assuming depth measurements being always strictly orthogonal to the skull surface. As mentioned in Section 3, this assumption is not always valid. A number of rules are thus used in traditional facial reconstruction to help locate certain features of the face based on the skull shape, employing empirical knowledge about shape relations between skin and skull [Taylor

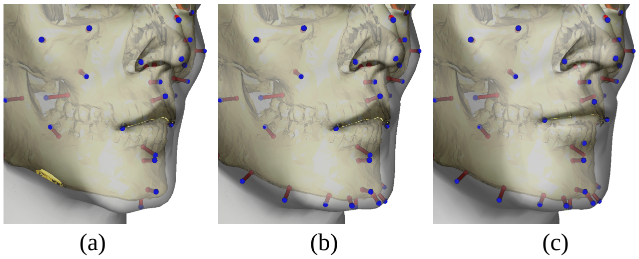


Figure 6: Fitting stages, shown on the lower face. a) Warp using only user-specified landmarks (some skull areas still intersecting the skin); b) with automatically interpolated landmarks on the mandible; c) using additional heuristics for lip and nose shaping.

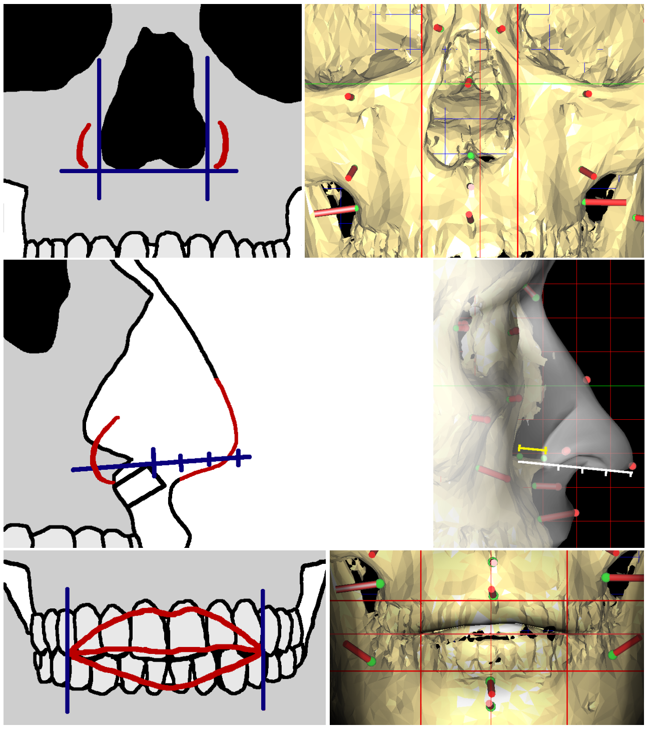


Figure 7: Comparison of heuristics used in traditional reconstruction (left) with our graphical interface (right). (Note: different skulls are used in the adjoining images.) Top: estimation of nose width; center: positioning of the nose tip; bottom: setting lip width, height, and mouth corner position.

2001]. We have translated some of these heuristics for use with the skull landmark editor: the final fitting result, as shown in Fig. 6(c), is obtained by including this additional user input.

To keep the user interface uniform, most rules are expressed by the placement of vertical and horizontal guides in a frontal view of the skull. From this user input, the placement of a few landmarks on the skin is adjusted, resulting in a new target landmark configuration. The updated landmark set is used to compute another warp function, which deforms the pre-fitted head model in the adjusted regions. Five rules influence the shape of the nose and the shape of the mouth, as shown in Fig. 7:

- The width of the nose wings corresponds to the width of the nasal aperture at its widest point, plus 5mm on either side in Caucasoids. In the editor, the user places two vertical guides to the left and right of the nasal aperture. From their position, the displacement of the two *al*[1] skin landmarks placed at the nose wings is computed (cf. Fig. 7, top row).

- The position of the nose tip depends on the shape of the anterior nasal spine. According to KROGMAN's formula [Taylor 2001, p. 443], the tip of the nose is in the extension of the nasal spine. Starting from the z value of the tissue depth marker directly below the nose (mid-philtrum, see Appendix A), the line is extended by three times the length of the nasal spine (cf. the white and yellow lines in the rightmost image of Fig. 7, middle row). In the editor, begin and end points of the nasal spine are marked. The *prn* landmark at the nose tip is then displaced according to the formula.

- The width of the mouth is determined by measuring the front six teeth, placing the mouth angles horizontally at the junction between the canine and the first premolar in a frontal view. Two vertical guides are used for positioning the *ch* landmarks located at the mouth angles (vertical lines in Fig. 7, bottom row).

- The thickness of the lips is determined by examining the upper and lower frontal teeth. Seen from the front, the transition between the lip and facial skin is placed at the transition between the enamel and the root part of the teeth. Two horizontal guides are placed by the user at the upper and lower transition, respectively. This determines the vertical position of the *id* and *sd* landmarks marking the lip boundary (top and bottom horizontal lines in Fig. 7, bottom row).

- The parting line between the lips is slightly above the blades of the incisors. This determines the vertical placement of the *ch* landmarks (middle horizontal line in Fig. 7, bottom row).

Using these heuristics, a better estimate of the mouth and nose shapes can be computed. The effect is strongest on the lip margins, since the assumption of an orthogonal connection between corresponding skin and skull landmarks is in fact not correct at these sites, as the right part of Fig. 4 shows. The initial deformation thus gives a good estimate of the tissue thickness of the lips while the second deformation using the information provided by interactive guide adjustment refines the vertical placement of the lip margins.

5 Facial Expressions and Rendering

In manual facial reconstruction, a neutral pose of the face is preferred as the most "generic" facial expression. Other expressions could be helpful for identification purposes, but the cost of modeling separate versions of the head model is prohibitive. In our virtual reconstruction approach, this does not pose a problem. Since the fitted head model has the animatable structure of skin and muscles, different facial expressions can be assumed by setting muscle contractions, as in other physics-based facial animation systems [Kähler et al. 2001; Lee et al. 1995]. Fig. 8 shows how muscles are used to form different facial expressions.

For a completely animatable head model, it is necessary to include a separately controllable mandible, a tongue, rotatable eyeballs, and eye lids into the head model. We have decidedly left them out of the reconstruction approach since these features are not particularly useful in this application: while a modest change of expression such as a smile or a frown might aid identification, rolling of eyes, blinking, and talking would probably not. It is also nearly impossible to correctly guess details such as a specific way of speaking—errors in this respect would produce rather misleading results in a real identification case. The effort of placing tongue, eye, and potentially teeth models thus does not offset the benefits.

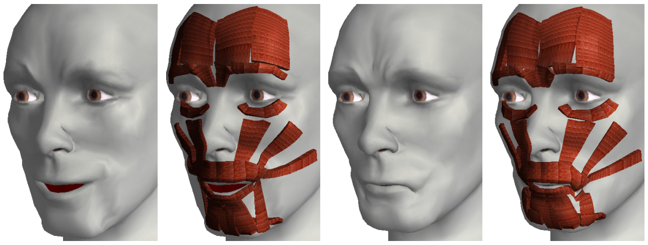


Figure 8: Expressions on the generic head model and the corresponding muscle configurations.

[1] see, e.g., [Farkas 1994] for a definition of standard facial landmarks

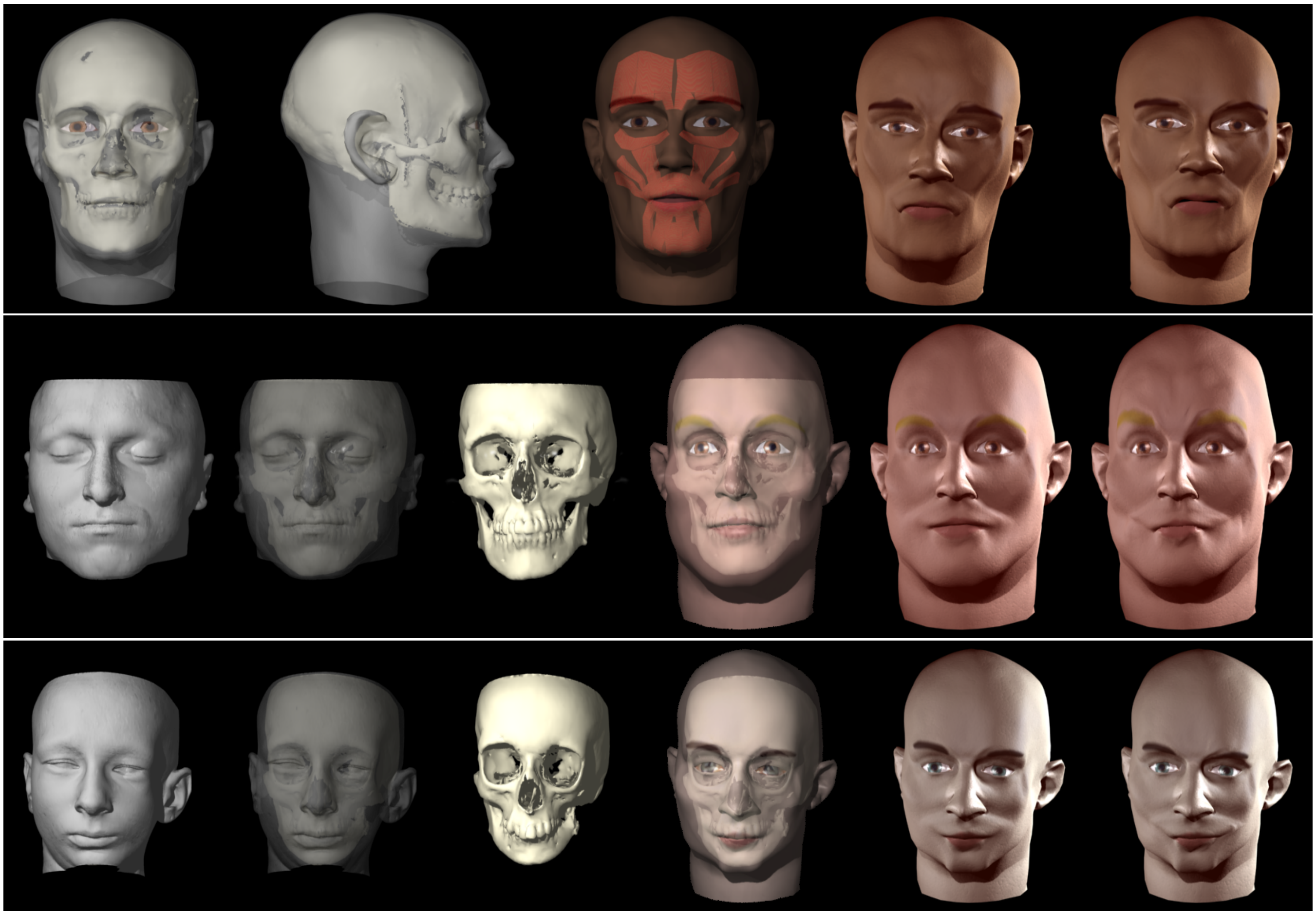


Figure 9: Examples of facial reconstructions created with our system. Top: model created from a scanned real skull, showing fit of skin to skull, transferred muscles, and two facial expressions. Middle: Reconstruction from a volume scan of a male, showing the actual face as contained in the data, superimpositions of the actual and the reconstructed face with the skull, and the reconstruction with neutral and "worried" expression. Bottom: Reconstruction from volume scan of a female with strong skull deformations. The CT data sets don't contain the top and bottom of the heads, thus the source skull and face models are cut off. The actual head height had to be guessed in these cases.

If additional information about the modeled person is available, for instance, from remnants of hair found with the skull, the resulting mesh can be colored correspondingly. Our system includes basic capabilities for coloring the parts associated with skin, lip, and eyebrows in the model's texture map. Colors can be adjusted interactively in HSV space on the reconstructed head model. Finally, the color adjustments are merged into a neutral base texture and saved as a new texture map. The fitted, texture-mapped triangle mesh can be easily imported into various rendering packages for display. The examples shown in Fig. 9 show three different skin colorations created in this way.

6 Results

We have tested our technique on a real skull that was made available to us by a forensic institute and on two medical volume scans. All data pertains to individuals of Caucasian type. Each reconstruction required approximately an hour of interactive work, excluding time for data acquisition.

The real skull, depicted on the first page of this paper, was unearthed on a construction site and belongs to an unidentified male, approximately 35 years of age. As can be seen from the hole in the frontal bone, he was killed by a head shot—the owner of this skull probably was a war victim or a soldier. After scanning the skull, the resulting mesh was simplified to 100k triangles. Interactive placement of skull landmarks and facial feature guides was relatively easy in this case since the skull is complete and in good condition. Due to its war-time origin, we assumed the face to be rather skinny, so we selected the "slender" tissue thickness table. Fitting results can be seen in Fig. 9, top row. Since the actual appearance of the individual is unknown, the accuracy of the reconstruction can only be guessed. Nonetheless, our reconstruction seems plausible. Notably, the shape of the chin, which can be predicted from the corresponding region on the skull, has been reproduced well.

To show examples utilizing other data sources, and also for validation, we extracted skull and skin surfaces from medical volume scans. The first data set, shown in the middle row of Fig. 9, pertains to a male subject of roughly 30 years. The subject's face is rather bulky, so we chose the "obese" tissue thickness data set (in a real case, this choice would have to be made based on other available information such as the size of clothes, if present). Our first reconstruction attempts showed a consistent emphasis on prominent cheek bones and hollow cheeks: no matter which data set we picked, the face would become more bulky, but not show the expected general roundness of the face. This effect is demonstrated in Fig. 10 on variations of our first model. A closer examination

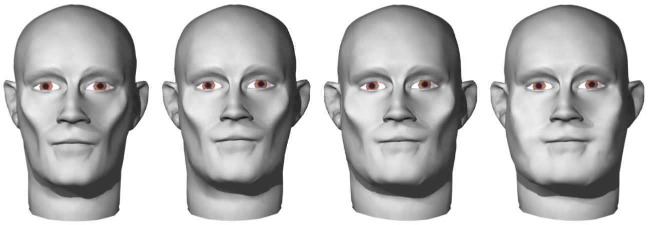


Figure 10: Left to right: RHINE's traditional "slender", "average", and "obese" tissue depth tables (cf. [Taylor 2001, p. 350 ff.]) often result in hollow cheeks and prominent cheekbones (see also Fig. 9). Rightmost image: the shape can be improved by "bulging out" the affected mesh areas.

revealed that the reason lies in the relatively low thickness values RHINE assigned to the landmarks defining the cheek region (*sbm2* and *spm2* in Table 1). After excluding these two landmarks, we obtained the results shown in Fig. 9. The rightmost image in Fig. 10 shows how simple mesh modeling techniques could be used at this point to improve and individualize the reconstruction.

The second volume data set shows a female patient with strong skull deformations. We produced a reconstruction of this face to test the method with a decidedly non-average skull shape. The result can be seen in the bottom row of Fig. 9. Since our automatic landmark interpolation scheme (see Section 4.2) is designed to handle the normal range of skull variations, the unusual shape of the mandible resulted in very sparse sampling of the chin area. Another prominent feature of the skull data is the protrusion of one incisor, pushing the upper lip to the front. We modeled this effect by moving the *sd* landmark a few millimeters down onto the blade of the incisor, thus pushing the associated skin landmark forward as well. This did not impair the positioning of the upper lip boundary since this is adjusted separately by the mouth guides (cf. Fig. 7).

7 Conclusion and Future Work

The face reconstruction approach presented in this paper mirrors the manual tissue depth method and thus has essentially the same prediction power. Our results show overall good reproduction of facial shape and proportions, and some surprisingly well-matched details. It should be noted that our examples were produced by computer scientists with no training in forensic reconstruction.

The advantages of the computerized solution are evident: instead of weeks, it takes less than a day to create a reconstructed face model, including scanning of the skull. Once the scan data is marked with landmarks, different varieties such as slimmer or more obese versions can be produced within seconds at the push of a button, which is practically impossible with the manual method due to the vast amount of time needed for production of a single model. Slight variations in facial expression can also be obtained quite easily by animating the muscle structure underlying the model.

Since the virtual reconstruction is based on 3D scans, which can be acquired contact-free, the risk of damage to the original skull is reduced. On the other hand, the scanning process has inherent limitations: depending on the maximum resolution of the digital scanner, much of the finer detail on the skull is lost. The delicate structure of, for instance, the nasal spine cannot be fully captured with current scanning technology. For this reason, it is necessary to consult the original skull from time to time for reference.

In our experiments, we often found that surface normals on the scanned skull geometry do not always behave the way they should, reflecting the orientation of the surface only very locally. It might be useful to consider an average of normals in a larger area around the landmark position to solve this. Sometimes, it would be desirable to adjust the orientation manually.

The interactive system allows for an iterative reconstruction approach: a model is produced quickly from a given landmark configuration, so landmarks can be edited repeatedly until the desired result is obtained. The emphasis on the interaction component makes the speed of the fitting process an important issue. While the actual calculation of the warp function and the deformation of the mesh are performed instantaneously, about five seconds are needed in our test setting on a 1.7 GHz Pentium Xeon to examine skull and skin for potential insertion of additional landmarks. This time is for the largest part used for ray intersections of the skull and skin meshes, which are done in a brute force manner. We expect a big speed-up through the use of space partitioning techniques.

For practical use, the facial reconstruction system should provide more editing facilities for skin details and hair. Useful additions include, for instance, a choice of templates for haircuts and facial features such as eyebrow shapes, beards, and wrinkles. At this point, large-scale validation of the system would be necessary to evaluate the usability of the system.

As TAYLOR writes in her book, the tissue depth values should not be taken at face value in three-dimensional facial reconstruction, but rather act as guides for the final facial reconstruction, which still relies heavily on artistic skills and intuition. Our tests confirm that strict adherance to RHINE's data for the solution of the interpolation problem is too limiting. This indicates not a weakness in our method, but reflects the low number of samples (between 3 and 37 in each group) and the technical limitations at the time RHINE assembled his data tables. Given the current state of technology, more samples of higher precision could be acquired, resulting in much more comprehensive and usable data. Ultimately, computer-based facial reconstruction could then even become superior to the traditional approach.

8 Acknowledgements

The authors would like to thank Dr. D. Buhmann from the Institute of Forensic Medicine, Saarland University, for his valuable comments and for providing the CT data sets.

References

ARCHER, K. M. 1997. *Craniofacial Reconstruction using hierarchical B-Spline Interpolation*. Master's thesis, University of British Columbia, Department of Electrical and Computer Engineering.

BLANZ, V., AND VETTER, T. 1999. A Morphable Model for the Synthesis of 3D Faces. In *Proc. ACM SIGGRAPH 1999*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 187–194.

BOOKSTEIN, F. L. 1997. *Morphometric Tools for Landmark Data*. Cambridge University Press.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and Representation of 3D Objects With Radial Basis Functions. ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 67–76.

CHADWICK, J. E., HAUMANN, D. R., AND PARENT, R. E. 1989. Layered Construction for Deformable Animated Characters. In *Computer Graphics (Proc. ACM SIGGRAPH 89)*, 243–252.

DUCHON, J. 1977. Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, W. Schempp and K. Zeller, Eds., vol. 571 of *Lecture Notes in Mathematics*, 85–100.

FARKAS, L. G., Ed. 1994. *Anthropometry of the Head and Face*, 2nd ed. Raven Press.

FEDOSYUTKIN, B. A., AND NAINYS, J. V. 1993. *Forensic Analysis of the Skull*. Wiley-Liss, ch. 15: The Relationship of Skull Morphology to Facial Features, 199–213.

Garland, M., and Heckbert, P. S. 1997. Surface simplification using quadric error metrics. In *SIGGRAPH 97 Conference Proceedings*, 209–216.

Grüner, O. 1993. *Forensic Analysis of the Skull*. Wiley-Liss, ch. 3: Identification of Skulls: A Historical Review and Practical Applications.

Helmer, R. P., Röhricht, S., Petersen, D., and Möhr, F. 1993. *Forensic Analysis of the Skull*. Wiley-Liss, ch. 17: Assessment of the Reliability of Facial Reconstruction, 229–246.

Kähler, K., Haber, J., and Seidel, H.-P. 2001. Geometry-based Muscle Modeling for Facial Animation. In *Proc. Graphics Interface 2001*, 37–46.

Kähler, K., Haber, J., Yamauchi, H., and Seidel, H.-P. 2002. Head shop: Generating animated head models with anatomical structure. In *ACM SIGGRAPH Symposium on Computer Animation*, ACM SIGGRAPH, 55–64.

Koch, R. M., Gross, M. H., and Bosshard, A. A. 1998. Emotion Editing using Finite Elements. In *Computer Graphics Forum (Proc. Eurographics '98)*, vol. 17, C295–C302.

Krogman, W. M. 1946. The reconstruction of the living head from the skull. *FBI Law Enforcement Bulletin* (July).

Lebedinskaya, G. V., Balueva, T. S., and Veselovskaya, E. V. 1993. *Forensic Analysis of the Skull*. Wiley-Liss, ch. 14: Principles of Facial Reconstruction, 183–198.

Lee, Y., Terzopoulos, D., and Waters, K. 1995. Realistic Modeling for Facial Animations. In *Proc. ACM SIGGRAPH 1995*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 55–62.

Michael, S., and Chen, M. 1996. The 3D reconstruction of facial features using volume distortion. In *Proc. 14th Eurographics UK Conference*, 297–305.

Miyasaka, S., Yoshino, M., Imaizumi, K., and Seta, S. 1995. The computer-aided facial reconstruction system. *Forensic Science Int. 74*, 1-2, 155–165.

Moore, W. J., and Lavelle, C. L. B. 1974. *Growth of the Facial Skeleton in the Hominoidea*. Academic Press, London.

Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., and Salesin, D. H. 1998. Synthesizing Realistic Facial Expressions from Photographs. In *Proc. ACM SIGGRAPH 1998*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 75–84.

Platt, S. M., and Badler, N. I. 1981. Animating Facial Expressions. In *Computer Graphics (Proc. ACM SIGGRAPH 81)*, 245–252.

Rhine, J. S., and Campbell, H. R. 1980. Thickness of facial tissues in American blacks. *Journal of Forensic Sciences 25*, 4, 847–858.

Rhine, J. S., and Moore, C. E. 1984. Tables of facial tissue thickness of American Caucasoids in forensic anthropology. *Maxwell Museum Technical Series 1*.

Scheepers, F., Parent, R. E., Carlson, W. E., and May, S. F. 1997. Anatomy-Based Modeling of the Human Musculature. In *Proc. ACM SIGGRAPH 1997*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 163–172.

Sederberg, T. W., and Parry, S. R. 1986. Free-Form Deformation of Solid Geometric Models. *Computer Graphics (Proc. ACM SIGGRAPH 86) 20*, 4 (Aug.), 151–160.

Szeliski, R., and Lavallée, S. 1996. Matching 3-D Anatomical Surfaces with Non-Rigid Deformations using Octree-Splines. *International Journal of Computer Vision 18*, 2, 171–186.

Taylor, K. T. 2001. *Forensic Art and Illustration*. CRC Press LLC.

Terzopoulos, D., and Waters, K. 1990. Physically-based Facial Modelling, Analysis, and Animation. *Journal of Visualization and Computer Animation 1*, 2 (Dec.), 73–80.

Vanezis, P., Vanezis, M., McCombe, G., and Niblett, T. 2000. Facial reconstruction using 3-D computer graphics. *Forensic Science Int. 108*, 2, 81–95.

Waters, K. 1987. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics (Proc. ACM SIGGRAPH 87)*, 17–24.

Weijs, W. A., and Hillen, B. 1986. Correlations between the cross-sectional area of the jaw muscles and craniofacial size and shape. *Am. J. Phys. Anthropol. 70*, 423–431.

Wilhelms, J., and Van Gelder, A. 1997. Anatomically Based Modeling. In *Proc. ACM SIGGRAPH 1997*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, 173–180.

y'Edynak, G. J., and İşcan, M. Y. 1993. *Forensic Analysis of the Skull*. Wiley-Liss, ch. 16: Anatomical and Artistic Guidelines for Forensic Facial Reconstruction, 215–227.

Zollikofer, C. P. E., Ponce De León, M. S., and Martin, R. D. 1998. Computer-Assisted Paleoanthropology. *Evolutionary Anthropology 6*, 2, 41–54.

A Landmark Set used for Reconstruction

Table 1 lists the paired landmarks on skin and skull that are used for the facial reconstruction approach described in this paper. Most skull landmark names and descriptions are taken from [Taylor 2001, page 350 ff.]. Short skull landmark names are listed in the *id* column. We have tried to adhere to naming conventions used in the forensic and anthropometric literature as much as possible [Taylor 2001; y'Edynak and İşcan 1993; Farkas 1994]. For simplicity, corresponding landmarks on skull and skin have the same short name in our system, which is not generally the case in the literature. In a few cases, marked by ∗ in the table, we invented short names. Not all skull landmarks have an "official" counterpart on the skin, so we placed the corresponding skin markers using our own judgment. The *mp* landmark pair is not part of the standard set. We added it to improve the alignment of skin to skull in the region behind the ears, where the mastoid process adds a bulge to the skull.

name	*id*	*description*
		Midline
Supraglabella	tr	Above glabella, identified with the hairline
Glabella	g	The most prominent point between the supraorbital ridges in the midsagittal plane
Nasion	n	The midpoint of the suture between the frontal and the two nasal bones
End of nasals	na	The anterior tip or the farthest point out on the nasal bones
Mid-philtrum	a	The mid line of the maxilla (east and west), placed as high as possible before the curvature of the anterior nasal spine begins
Upper lip margin (Supradentale)	sd	Centered between the maxillary (upper) central incisors at the level of the Cementum Enamel Junction (CEJ)
Lower lip margin (Infradentale)	id	Centered between the mandibula (lower) central incisors at the level of the Cementum Enamel Junction (CEJ)
Chin-lip fold (Supramentale)	b	The deepest mid line point of indentation on the mandible between the teeth and the chin protrusion
Mental eminence (Pogonion)	pog	The most anterior or projecting point in the mid line on the chin
Beneath chin (Menton)	me	The lowest point on the mandible
		Bilateral
Frontal eminence	fe*	Place on the projections at both sides of the forehead
Supraorbital	sci	Above the orbit, centered on the upper most margin or border
Suborbital	or	Below the orbit, centered on the lower most margin or border
Endocanthion	en	point at the inner commissure of the eye fissure; the landmark on the skin is slightly lateral to the one on the bone
Exocanthion	ex	point at the outer commissure of the eye fissure; the landmark on the skin is slightly medial to the one on the bone
Inferior malar	im	The lower portion of the maxilla, still on the cheekbone
Lateral orbit	lo	Drop a line from the outer margin of the orbit and place the marker about 10 mm below the orbit
Zygomatic arch, midway	zy	Halfway along the zygomatic arch (generally the most projecting point on the arch when viewed from above)
Supraglenoid	sg	above and slightly forward of the external auditory meatus
Gonion	go	The most lateral point on the mandibular angle
Supra M^2	spm2*	Above the second maxillary molar
Occlusal line	ol	On the mandible in alignment with the line where the teeth occlude or bite
Sub M_2	sbm2*	Below the second mandibular molar
Mastoid process	mp*	Most lateral part on the mastoid process behind and below the ear canal

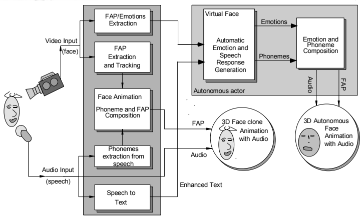
Table 1: Landmark set used for face reconstruction.

Speech Animation

Nadia Magnenat-Thalmann
MIRALab, University of Geneva
thalmann@miralab.unige.ch

Where do we stand today?

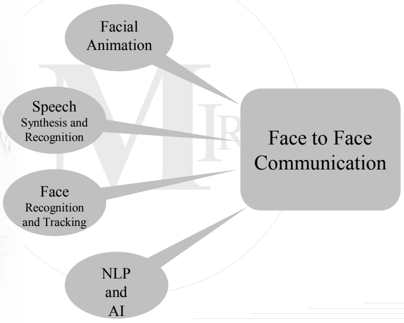
Face to Virtual Face Communication


Video Input
(face)
FAP/Emotions Extraction
FAP Extraction and Tracking
Face Animation
Phoneme and FAP Composition
Phonemes extraction from speech
Audio Input
(speech)
Speech to Text
Virtual Face
Automatic Emotion and Speech Response Generation
Emotions
Phonemes
Emotion and Phoneme Composition
Audio
FAP
Autonomous actor
FAP
3D Face clone Animation with Audio
Audio
3D Autonomous Face Animation with Audio
Enhanced Text

- •What are technologies?
- •What is the progress?
- •What is still absent?
- •What is the future?

Magnenat Thalmann N., Kalra P., Pandzic I.S., "**Direct Face-to-Face Communication Between Real and Virtual Humans**", International Journal of Information Technology, Vol.1, No.2, 1995, pp.145-157.

What are the technologies?



Facial Animation

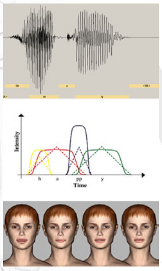
Speech Synthesis and Recognition

Face Recognition and Tracking

NLP and AI

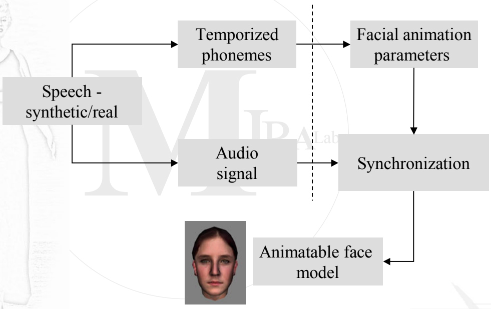
Face to Face Communication

Speech Animation : Hierarchy



Step	Technology	Methods
Temporized phonemes from speech (synthetic or real)	Phoneme recognition	Manual, semi-automatic or automatic
Phoneme transition	Co-articulation	Rules based, automatic
Viseme generation and animation	Viseme definition, Synchronization with sound	Automatic

Animated Talking Heads – a typical system



Speech - synthetic/real

Temporized phonemes

Audio signal

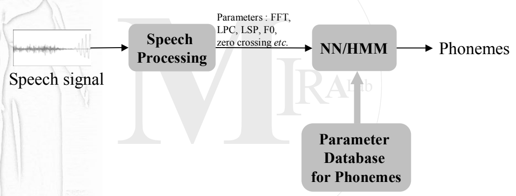
Facial animation parameters

Synchronization

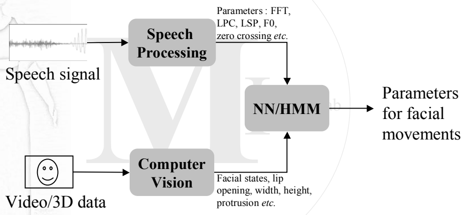
Animatable face model

Speech Animation from Natural Voice (for cloned avatars)

(Only) Acoustic Analysis


Speech signal
Speech Processing
Parameters : FFT, LPC, LSP, F0, zero crossing *etc.*
NN/HMM
Phonemes
Parameter Database for Phonemes

Acoustic and Visual Analysis


Speech signal
Speech Processing
Parameters : FFT, LPC, LSP, F0, zero crossing *etc.*
NN/HMM
Parameters for facial movements
Video/3D data
Computer Vision
Facial states, lip opening, width, height, protrusion *etc.*

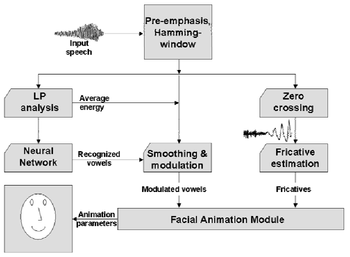
Comparison

Acoustic Analysis	Acoustic-Visual Analysis
The output "phonemes", suitable for any facial animation system	Output parameters (facial states/lip width, height *etc.*) are tied to a particular animation system
Resulting facial animation not affected by training database	Resulting facial animation closely affected by the training database
Ease in training data collection (only speech)	Training data is synchronized speech and video/3D capture
Only lip/mouth movements can be generated	Technique can be used for synthesis of other facial movements (eyebrow, nods)
Co-articulation model needs to be applied to resulting phonemes	Co-articulation effect is inherently taken care of in analysis
Greater language dependence	Less language dependence

Challenges

- Independent of language and speaker
- Independent of face model used for animation
- Minimal training requirements
- Simplicity of tools, algorithm and implementation

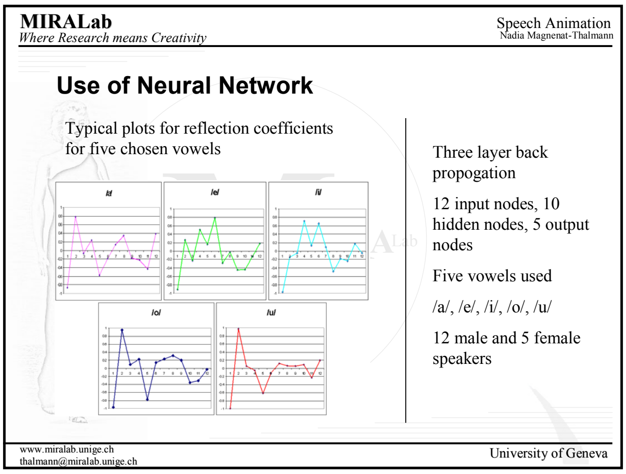
Speech Driven Talking Head: an example


Input speech
Pre-emphasis, Hamming-window
LP analysis
Average energy
Zero crossing
Neural Network
Recognized vowels
Smoothing & modulation
Fricative estimation
Modulated vowels
Fricatives
Animation parameters
Facial Animation Module

Speech Analysis

Parameter extraction

- Choice of LP Analysis
  - LP derived reflection coefficients are directly related to vocal tract shape [Wakita]
  - Phonemes can be characterized by vocal tract shape
- Limitations
  - Works well for vowels so we choose the most common five vowels /a/, /e/, /i/, /o/, /u/.
  - For the consonants?



Use of Neural Network

Typical plots for reflection coefficients for five chosen vowels

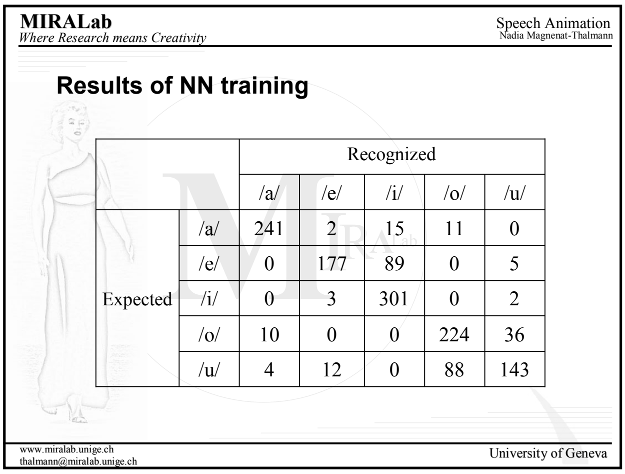
Three layer back propogation

12 input nodes, 10 hidden nodes, 5 output nodes

Five vowels used

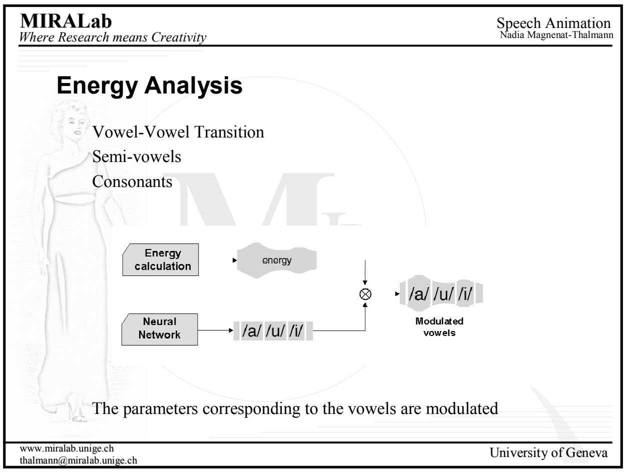
/a/, /e/, /i/, /o/, /u/

12 male and 5 female speakers



Results of NN training

		Recognized				
		/a/	/e/	/i/	/o/	/u/
Expected	/a/	241	2	15	11	0
	/e/	0	177	89	0	5
	/i/	0	3	301	0	2
	/o/	10	0	0	224	36
	/u/	4	12	0	88	143



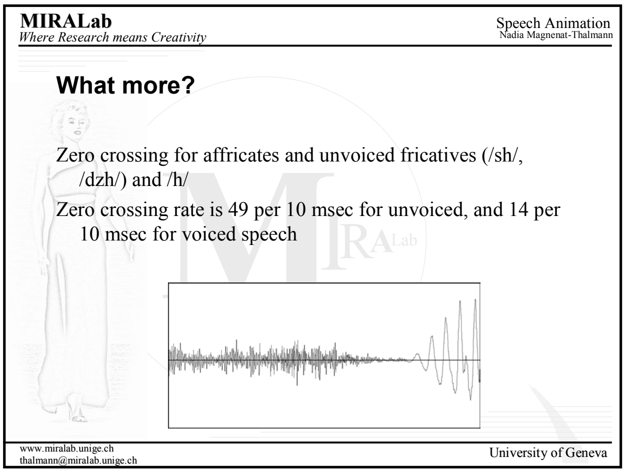
Energy Analysis

Vowel-Vowel Transition

Semi-vowels

Consonants

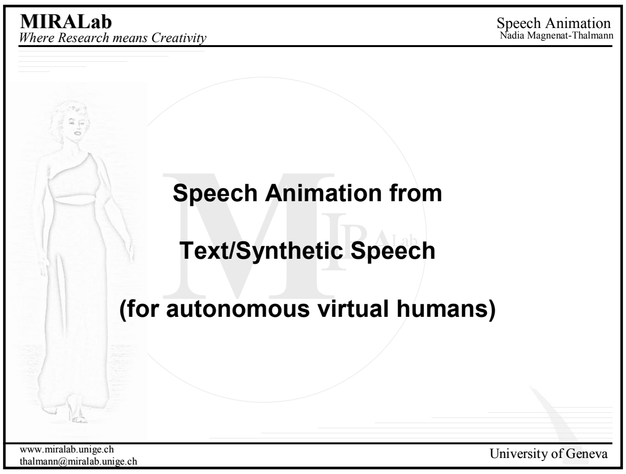
The parameters corresponding to the vowels are modulated



What more?

Zero crossing for affricates and unvoiced fricatives (/sh/, /dzh/) and /h/

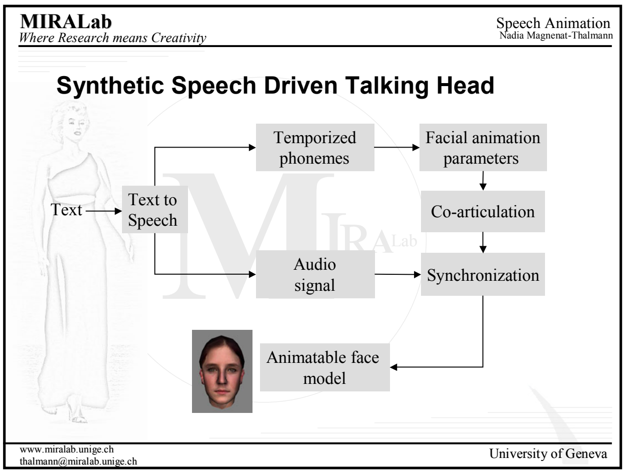
Zero crossing rate is 49 per 10 msec for unvoiced, and 14 per 10 msec for voiced speech



Speech Animation from

Text/Synthetic Speech

(for autonomous virtual humans)

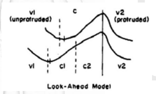


Synthetic Speech Driven Talking Head

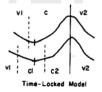
Speech Co-articulation

Co-articulation is a phenomenon observed during fluent speech, in which facial movements corresponding to one phonetic or visemic segments are influenced by those corresponding to the neighboring segments.

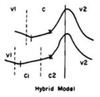
Example: a V1-C-V2 sequence where V1 is un-protruded (eg. ‘a’) and V2 is protruded (eg. ‘u’)



Transition towards v2 starts as soon as v1 ends



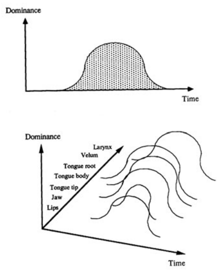
Transition towards v2 starts a fixed time interval before v2 begins



Transition towards v2 takes place in two phases

M. M. Cohen, D.W. Massaro, "Modelling coarticulation in synthetic visual speech", in N. M. Thalmann and D. Thalmann, *Models and techniques in Computer Animation*, Spinger-Verlag, 1993, pp. 139-156.

Articulatory Gesture Model



- •Each speech segment (typically a viseme) has dominance that increases and decreases over time
- •Adjacent visemes have overlapping dominance functions that will blend over time
- •Each viseme may have a different dominance function for each articulator

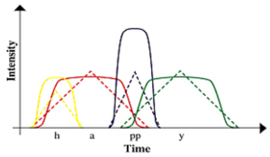
A. Löfqvist, “Speech as audible gestures”, in *Speech Production and Speech Modeling, Kluwer Academic Publishers, 289-322*

Co-articulation Models for Talking Head

Pelachaud (1991) :
“Look ahead” model based on deformability of phonemes
Also considered muscle contraction times

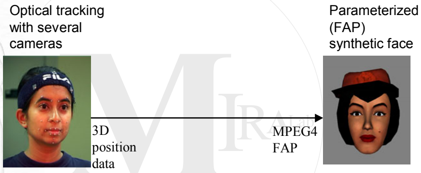
Cohen & Massaro (1992) :
Non-linear dominance and blending functions designed for each phoneme

In Summary



Define weight (dominance), and overlap according to phoneme group.

Performance Driven Facial Animation



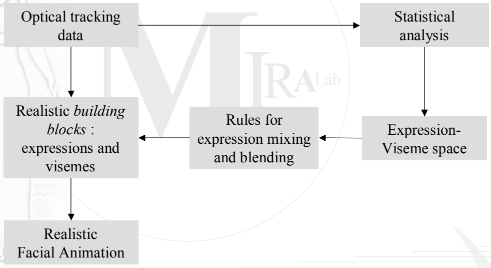
Enhances realism to a great degree

Enables design of the *building blocks*

Limitations : complex equipment, availability of skilled performer

Realism in Talking Heads

Can we combine *flexibility* of facial animation design and *realism* of performance driven facial animation? How?



Optical tracking data

Statistical analysis

Realistic *building blocks* : expressions and visemes

Rules for expression mixing and blending

Expression-Viseme space

Realistic Facial Animation

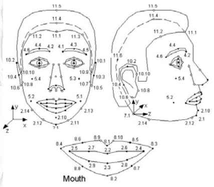
What is PCA

PCA is a well-known multivariate statistical analysis technique aimed at :

- reducing the dimensionality of a dataset, which consists of a large number of interrelated variables
- retaining as much as possible of the variation present in the dataset
- transforming the existing dataset into a new set of variables called the principal components (PC)

The PCs are uncorrelated and are ordered so that the first few PCs retain the most of the variation present in all of the original dataset.

Why PCA



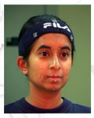
For facial capture

High correlation between facial feature points

Large amount of capture data for speech

Capturing individual as well as collective movement dynamics important during expressive speech

Data Capture



Optical Tracking system : Vicon

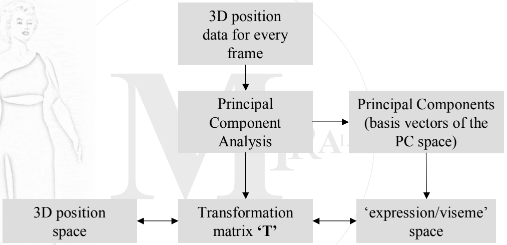
27 optical markers, 6 Cameras

Extraction of 3D positions of markers

100 phoneme rich sentences from TIMIT database

3D position data of 14 markers around lips and cheeks used for PCA

Data Analysis



3D position data for every frame

Principal Component Analysis

Principal Components (basis vectors of the PC space)

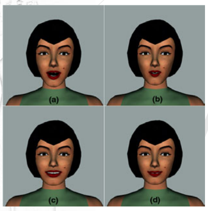
3D position space

Transformation matrix **'T'**

'expression/viseme' space

Analysis results into a transformation between 3D position space and the newly constructed expression/viseme space

What are the Principal Components



(a) Open mouth
(b) Lip protrusion
(c) Lip sucking
(d) Raise cornerlips

The facial movements are controlled by single parameters, as opposed to several MPEG4 parameters needed to control the same facial movement

Eg. 'Open Mouth ' affects not only lips, but jaw and cheek region also

Thus the Principal Components take care of global facial movements using minimum number of parameters and provide higher level parameterization for facial animation design

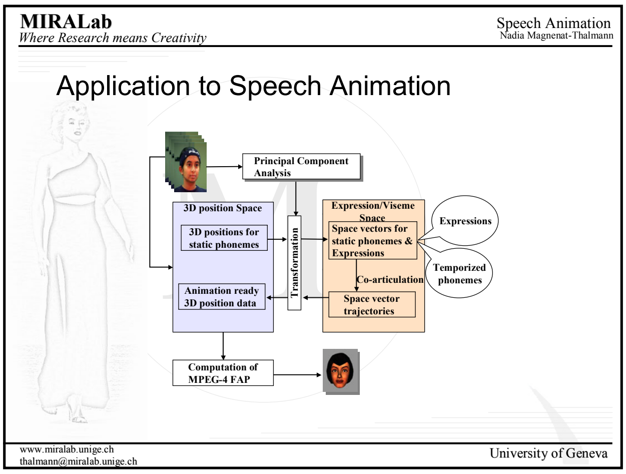
Expression and Viseme Space

•The 'Principal Components' form the basis or the 'principal axes' of the abstract *Expression and Viseme* space

•Each point in the *Expression and Viseme* space is a facial expression, a viseme, or a combination

•Transition in this space from one point (expression) to another, results in smooth and realistic transition in the 3D position space giving a new way of achieving keyframe animations.

•A combination of points in this space results in realistic blending and combination of visemes and expressions in 3D position space, and hence a realistic expressive speech animation.



Application to Speech Animation

Principal Component Analysis

3D position Space

3D positions for static phonemes

Animation ready 3D position data

Transformation

Expression/Viseme Space

Space vectors for static phonemes & Expressions

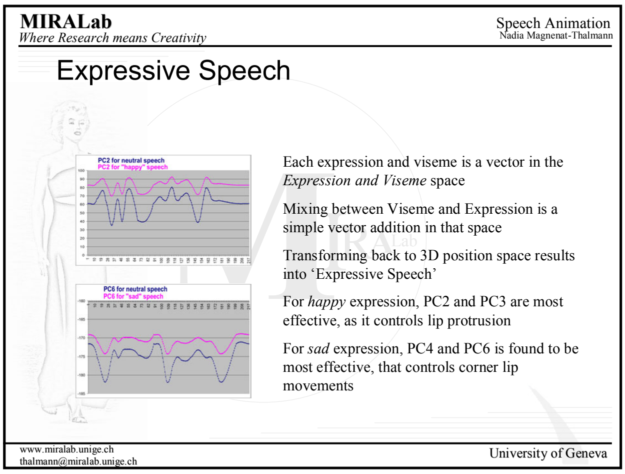
Co-articulation

Space vector trajectories

Expressions

Temporized phonemes

Computation of MPEG-4 FAP



Expressive Speech

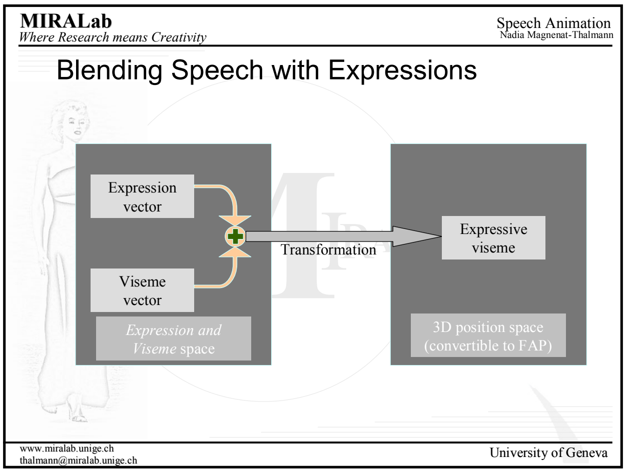
Each expression and viseme is a vector in the *Expression and Viseme* space

Mixing between Viseme and Expression is a simple vector addition in that space

Transforming back to 3D position space results into 'Expressive Speech'

For *happy* expression, PC2 and PC3 are most effective, as it controls lip protrusion

For *sad* expression, PC4 and PC6 is found to be most effective, that controls corner lip movements



Blending Speech with Expressions

Expression vector

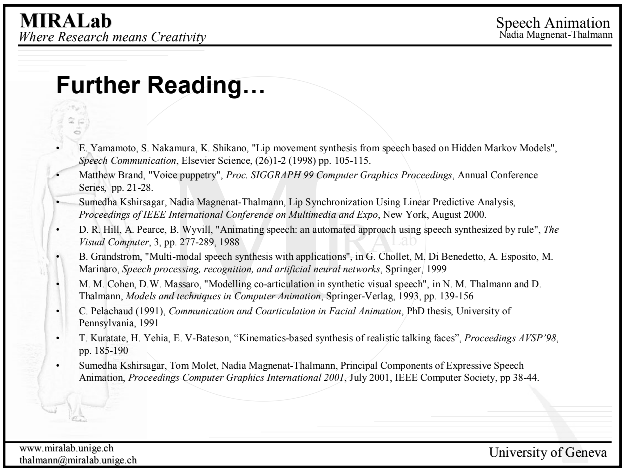
Viseme vector

Transformation

Expressive viseme

Expression and Viseme space

3D position space (convertible to FAP)



Further Reading…

- E. Yamamoto, S. Nakamura, K. Shikano, "Lip movement synthesis from speech based on Hidden Markov Models", *Speech Communication*, Elsevier Science, (26)1-2 (1998) pp. 105-115.
- Matthew Brand, "Voice puppetry", *Proc. SIGGRAPH 99 Computer Graphics Proceedings*, Annual Conference Series, pp. 21-28.
- Sumedha Kshirsagar, Nadia Magnenat-Thalmann, Lip Synchronization Using Linear Predictive Analysis, *Proceedings of IEEE International Conference on Multimedia and Expo*, New York, August 2000.
- D. R. Hill, A. Pearce, B. Wyvill, "Animating speech: an automated approach using speech synthesized by rule", *The Visual Computer*, 3, pp. 277-289, 1988
- B. Grandstrom, "Multi-modal speech synthesis with applications", in G. Chollet, M. Di Benedetto, A. Esposito, M. Marinaro, *Speech processing, recognition, and artificial neural networks*, Springer, 1999
- M. M. Cohen, D.W. Massaro, "Modelling co-articulation in synthetic visual speech", in N. M. Thalmann and D. Thalmann, *Models and techniques in Computer Animation*, Springer-Verlag, 1993, pp. 139-156
- C. Pelachaud (1991), *Communication and Coarticulation in Facial Animation*, PhD thesis, University of Pennsylvania, 1991
- T. Kuratate, H. Yehia, E. V-Bateson, "Kinematics-based synthesis of realistic talking faces", *Proceedings AVSP'98*, pp. 185-190
- Sumedha Kshirsagar, Tom Molet, Nadia Magnenat-Thalmann, Principal Components of Expressive Speech Animation, *Proceedings Computer Graphics International 2001*, July 2001, IEEE Computer Society, pp 38-44.

Lip Synchronization Using Linear Predictive Analysis

Sumedha Kshirsagar, Nadia Magnenat-Thalmann
MIRALAB, CUI, University of Geneva
24 rue de General Dufour
CH-1211 Geneve, SWITZERLAND
+41-22-7057769
{sumedha.kshirsagar,nadia.thalmann}@cui.unige.ch

ABSTRACT

Linear Predictive analysis is a widely used technique for speech analysis and encoding. In this paper, we discuss the issues involved in its application to phoneme extraction and lip synchronization. The LP analysis results in a set of *reflection coefficients* that are closely related to the vocal tract shape. Since the vocal tract shape can be correlated with the phoneme being spoken, LP analysis can be directly applied to phoneme extraction. We use neural networks to train and classify the reflection coefficients into a set of vowels. In addition, average energy is used to take care of vowel-vowel and vowel-consonant transitions, whereas the zero crossing information is used to detect the presence of fricatives. We directly apply the extracted phoneme information to our synthetic 3D face model. The proposed method is fast, easy to implement, and adequate for real time speech animation. As the method does not rely on language structure or speech recognition, it is language independent. Moreover, the method is speaker independent. It can be applied to lip synchronization for entertainment applications and *avatar* animation in virtual environments.

Keywords

LP analysis, lip synchronization, real-time speech animation

1. INTRODUCTION

In today's multi-modal user interactive systems, talking heads form an important and essential part. For virtual presenters, storytellers, and *avatars* in virtual environments, synthetic faces talking in natural voice are gaining more potential. The advances in speech synthesis technologies are resulting in better quality computer generated voices. Nevertheless, using natural voice for the animation of synthetic faces remains a challenging area of research in computer animation. The problem can be easily divided into two parts; *viz.* extracting the mouth shape information from speech signal and then applying it to a synthetic 3D face model with synchronization for realistic animation. We concentrate on the former part in this paper, and briefly discuss the issues involved in the later.

The goal is to extract the parameters from speech signal which are directly or indirectly related to the mouth/lip movements. McAllister *et al* [1] used mouth shape descriptors called *moments* computed from the FFT coefficients of the speech signal as these parameters. LPC derived cepstral coefficients were used by Curinga, Lavagetto, and Vignoli [2]. They trained *Time Delay Neural Network* to take care of the co-articulation. Yamamoto, Nakamura, Shikano [3] and Tamura *et al* [4] used HMM techniques for the synthesis of lip movements from the speech signal. Morishima [5] described a real time voice driven talking head and its application to entertainment. He used LPC derived cepstral coefficients to extract mouth shape parameters using neural networks.

Most of the above mentioned researchers used the mouth shape parameters like width, height, lip-to-lip distance or the control point locations around the lips. These parameters were extracted from the video sequences associated with the speech recordings, and were then used for training. We propose to take a different approach that will allow us to apply the results to any generalized 3D head. The straightforward alternative is to extract the phonemes or *visemes* (visual counterparts of phonemes) directly from the speech signal. We choose LP analysis to extract the parameters from the speech signal. This technique, as explained in the subsequent sections, is inadequate for the consonants. Thus, it has limited use for the accurate phoneme extraction. We partly overcome the limitation by augmenting the results of vowel recognition with the energy envelope modulation. We also use zero crossing rate to recognize unvoiced fricatives. Our face model can directly animate such modulated vowels, making the animation process easy.

The next section explains the proposed system in brief. Section 3 explains the use of the LP analysis and use of neural networks, energy criterion and zero crossing rate for phoneme extraction. The issues related to 3D synthetic faces for speech animation are discussed in section 4. Finally, we give conclusions and discuss future work.

2. SYSTEM OVERVIEW

Figure 1 shows the overall block diagram of the system. Input speech is sampled at 10 kHz with a frame size of 20 ms. Preprocessing includes pre-emphasis and hamming windowing of the signal. Currently, no filtering is done for noise reduction. 12 reflection coefficients are calculated as a result of LP analysis. The coefficients are obtained from sustained vowel data and are used to train the neural network. As a result, one of the 5 chosen vowels (/a/, /e/, /i/, /o/, /u/) is obtained for the frame. We have chosen these vowels since we notice that the vowels in many languages can be roughly classified into these basic sounds or their combinations/variations. We use median filtering to smooth the resulting recognized vowels. The average energy of the signal is calculated as the zeroth auto-correlation coefficient over the frame and is used to decide the intensity of the detected vowel. Zero crossings are calculated to decide the presence of the unvoiced fricatives and affricates (/sh/, /ch/, /zh/ *etc.*). Finally, the Facial Animation Module generates the *Facial Animation Parameters* (FAP) as supported by MPEG-4 standard depending upon the phoneme input. Note that any 3D parameterized facial model can be used here. The speech animation is then easy using the pre-defined parameters for these extracted phonemes.

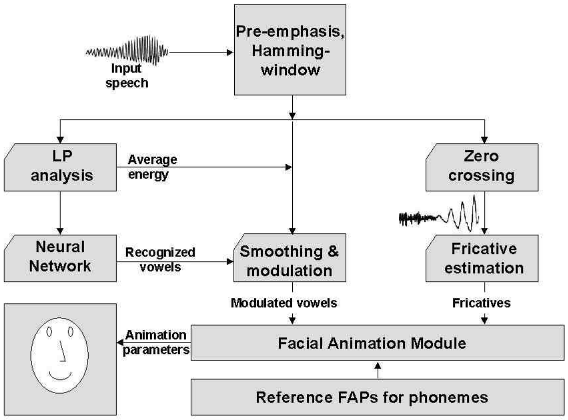


Figure 1: System overview

3. SPEECH ANLYSIS

As previously mentioned, several parameters can be used to correlate the speech signal to the mouth shape. Lewis and Parke [6] suggested the use of linear prediction for lip synchronization. However, they used Fourier transform of the negated zero extended LP coefficients for analysis. An analyzed speech frame was then classified using the *Euclidean distance norm* after comparing it with the spectra of reference phonemes. We use the LP derived *reflection coefficients*, the average energy in the speech signal and the zero crossing rate. This section explains the choice of these parameters in details.

3.1 Speech Production and LP Analysis

The human speech production system can be easily divided into the glottis or vocal cords and the vocal tract (mouth, tongue and lips). The glottal excitation acts as the source signal. The vocal tract, acting as a filter, then shapes it to generate the output speech. The phonemes can be characterized together by the excitation and the vocal tract shape. We concentrate on the vocal tract shape here. This subsection focuses on extraction of vowels, whereas issues involved in the extraction of some of the consonants are discussed in the subsequent subsections. For the production of vowels, the vocal tract shape is constant with time and uniform (without constrictions), with the sustained vibrations of the vocal cords.

Thus for the vowels, the vocal tract can be approximately modeled as a concatenation of a number of cylindrical tubes of uniform cross-sectional area [7]. Figure 2 shows a simple approximation of the model consisting of *m* acoustic tubes. The tubes have cross-sectional areas A_1 to A_m. Though these values have great variation from person to person, the relative distribution is similar for a given vowel. We are interested in extracting this vocal tract shape information, which will be directly useful for speech animation.

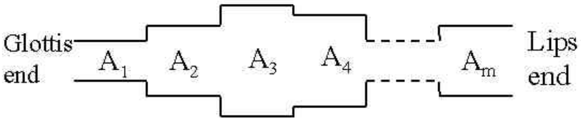


Figure 2: Schematic representation of vocal system and concatenated tube approximation

Wakita [8] compared the above acoustic filter model represented by the concatenated tubes, with the speech production model suggested by LP analysis. The details are beyond the scope of discussion here, but we state the result. The comparison between the acoustic tube model, and the LP derived model led to the following conclusion. The reflection coefficients r_l, computed as a by-product of the recursive LP algorithm, are directly related to the vocal tract area as per the concatenated tube model by the following equation.

$$r_l = \frac{A_{l-1} - A_l}{A_{l-1} + A_l}$$

As clearly depicted by the equation, the reflection coefficients are directly related to the variation of the vocal tract area for sustained vowels. A definite pattern observed in these coefficients for a particular vowel suggests the use of neural networks for classification.

3.2 Use of Neural Network

With the background given in the last subsection, the problem of recognizing the vowels reduces to a classification problem. A three-layer back-propagation neural network is widely used for a variety of pattern recognition and classification problems [9]. We use the same configuration to classify the reflection coefficients. There are 12 input nodes for the coefficients, 10 hidden nodes and 5 output nodes for the vowels. These parameters were tuned by running the training sessions several times on the data and studying the classification result. We train the network in five repeated cycles, every time using the data in a different random order. We use reflection coefficients from sustained vowel data and also short vowel segments extracted from continuous speech. The speech data was recorded from 12 male and 5 female speakers. The following table shows the classification results on the test data set consisting of 4 male and 3 female speakers. The utterances were chosen from sustained vowels and the frames were chosen randomly. Note the mis-recognition between /e/ and /i/, and /o/ and /u/. The mouth shapes for these pairs of vowels are also similar.

		Recognized				
		/a/	/e/	/i/	/o/	/u/
Expected	/a/	241	2	15	11	0
	/e/	0	177	89	0	5
	/i/	0	3	301	0	2
	/o/	10	0	0	224	36
	/u/	4	12	0	88	143

Table 1: Results of neural network classification

3.3 Energy Analysis

In the previous subsections, we have explained how the vowels can be extracted directly from the speech signal using LP analysis and neural networks. However, we are aware that the application of the vowels alone for speech animation is not sufficient. The vowel-to-vowel transition and the consonant information are missing, which are very important for realistic speech animation. The consonants are typically produced by creating a constriction at some place along the length of the vocal tract. During such constrictions/closures, the energy in the speech signal diminishes. Hence, we use the average energy in a speech frame to modulate the recognized vowel. It is calculated as the zeroth autocorrelation coefficient of the frame, and has already been computed during the LP analysis phase. Thus, the calculation of energy does not cause any additional computational overhead.

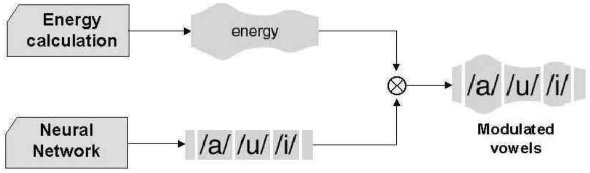


Figure 3: Modulating vowels with energy envelope

As an initialization process, we record background noise in the room to set the energy threshold for silence. Also, we record sustained vowel /a/ from the user asking her to say the utterance with maximum volume expected in normal speech. This enables us to compute the maximum energy. This value is used to get a normalized weighting factor for the vowels during normal speech.

As explained in Section 4, the parameterized face model (MPEG4 model in our case) enables us to animate these modulated vowels. The normalized weighting factor directly proportional to the energy in the speech signal is used to scale the parameters for the corresponding vowel. Figure 3 pictorially depicts the idea behind modulating vowels with energy envelope.

3.4 Zero Crossing

Using the energy content of the signal may result in false closure of mouth, especially in the case of affricates and unvoiced fricatives. For such cases, we can use the average zero crossing rate in the speech signal for each frame. The mean short time average zero crossing rate is 49 per 10 msec for unvoiced, and 14 per 10 msec for voiced speech [7]. This criterion is useful in making a distinction and is sufficient for our purpose. A short segment of the utterance /sh/ (as in sharp) shown in figure 4 highlights this criterion. In case of the presence of low energy in the speech frame, the zero crossing criterion decides the phoneme.

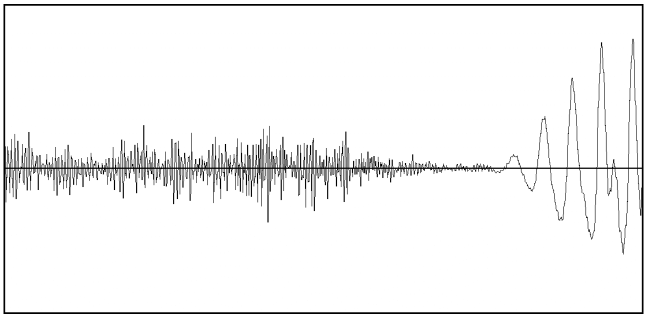


Figure 4: High zero crossing rate for fricative /sh/

4. FACIAL ANIMATION

So far, we have focused our attention on the extraction of phonemes directly from speech signal. In this section, we briefly consider the issues involved in the speech animation using this extracted information.

The phonemes extracted using the method described so far can be applied to any parameterized face model for speech animation. It is necessary to define the mouth shapes for the static phonemes in terms of these parameters. Depending upon the phoneme intensities, the corresponding parameter intensities can be set to achieve the speech animation. MPEG-4 standard provides such a way. The *Facial Animation Parameters* defined in the standard enable the user to define any facial expression in terms of these parameters with corresponding intensities. Moreover, since these parameters are normalized with respect to the distance between certain key feature points on the face, the animation results are consistent when applied to any face model. For more detail discussion on the MPEG-4 standard and MPEG-4 compatible 3D faces, refer to [10][11].

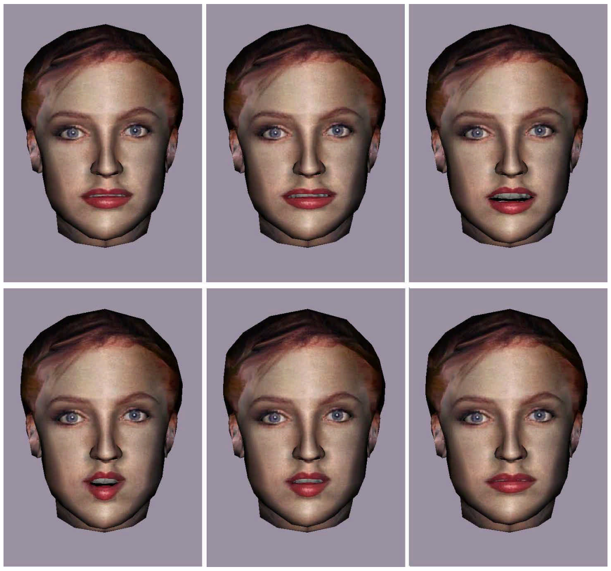


Figure 5: Frames of speech animation "hello"

In a networked virtual environment, an avatar of an individual can be truly represented by the individualized 3D face model and the voice of that individual using the lip synchronization method proposed here. An MPEG-4 compatible individualized 3D face of a person can be generated from two orthogonal photographs. We have used the 3D face models developed by [12]. We define the FAPs corresponding to the phonemes under consideration, and according to the modulation, the intensities of these FAPs are set to generate the facial animation in real time. Figure 5 shows successive frames for speech animation for the word "hello" which involves modulated vowels /a/, /e/, and /o/.

5. CONCLUSIONS AND FUTURE WORK

We have proposed a simple and fast method for realistic speech animation. As we are extracting higher level information (phonemes) directly from the speech signal, the results can be easily applied to any parameterized face model. We have used MPEG-4 compatible face model. The results of the speech animation using the recorded speech of different speakers can be seen at the following website: http://www.miralab.unige.ch/~sumedha/lipsynchronization. Note that the method is language as well as speaker independent. The score of the recognition given by the neural network can be used to combine two vowels generating a mouth shape that will represent the transition between them. We are aware that this method does not give accurate results as far as phoneme recognition is concerned. However, in the context of talking heads used for real time interactive system, the animation is satisfactory.

6. ACKNOWLEDGEMENTS

This work is supported by the EU ACTS VPARK project. We are thankful to the staff of the MIRALab for their valuable help in various matters.

7. REFERENCES

[1] D. V. McAllister, R. D. Rodman, D. L. Bitzer, A. S. Freeman, "Lip synchronization for Animation", *Proc. SIGGRAPH 97*, Los Angeles, CA, August 1997.

[2] Sergio Curinga, Fabio Lavagetto, Fabio Vignoli, "Lip movements synthesis using time delay neural networks", *Proc. EUSIPCO 96*, Sep. 1996.

[3] E. Yamamoto, S. Nakamura, K. Shikano, "Lip movement synthesis from speech based on Hidden Markov Models", *Speech Communication*, Elsevier Science, (26)1-2 (1998) pp. 105-115.

[4] M. Tamura, T. Masuko, T. Kobayashi, K. Tokuda, "Visual speech synthesis based in parameter generation from HMM : Speech driven and text-and-speech driven approaches", *Proc. AVSP 98*, International Conference on Auditory-Visual Speech Processing.

[5] S. Morishima, "Real-time talking head driven by voice and its application to communication and entertainment", *Proc. AVSP 98*, International Conference on Auditory-Visual Speech Processing.

[6] J. P. Lewis, F. I. Parke, "Automated lip-synch and speech synthesis for character animation", SIGGRAPH 1990, *Course Notes*, August 1990, pp.83-87.

[7] L. R. Rabiner, R. W. Schafer, *Digital Processing of Speech Signal*, Englewood Cliffs, New Jercy : Prentice Hall, 1978.

[8] Hisashi Wakita, "Direct estimation of the vocal tract shape by inverse filtering of the acoustic speech waveforms", *IEEE Trans. Audio & Electrocoustics*, Vol. 21, October 1973, pp. 417-427.

[9] J. A. Freeman, D. M. Skapura, "Neural Networks Algorithms, Applications, and Programming Techniques" Addison-Wesley, 1991.

[10] P. Doenges, F. Lavagetto, J. Ostermann, I. Pandzic, E. Petajan, "MPEG-4: Audio/Video and Synthetic Graphics/Audio for Mixed Media", Image Communications Journal, Vol.5, No.4, 1997, pp.433-463.

[11] M. Escher, I. Pandzic, N. M.-Thalmann, "Facial deformation from MPEG-4", *Proc. Computer Animation 98*, IEEE Computer Society, pp. 56-62.

[12] W. S. Lee, M. Escher, G. Sannier, N. M.-Thalmann, "MPEG-4 compatible faces from orthogonal photos", *Proc. Computer Animation 99*, IEEE Computer Society, pp. 186-194.

EUROGRAPHICS 2003

Textures

Jörg Haber

Textures are...

- a cheap means of conveying realism
- a tool for LoD management
- available both on graphics hardware and in modeling / rendering software
- useful for many rendering “tricks”

How to create textures from input images?

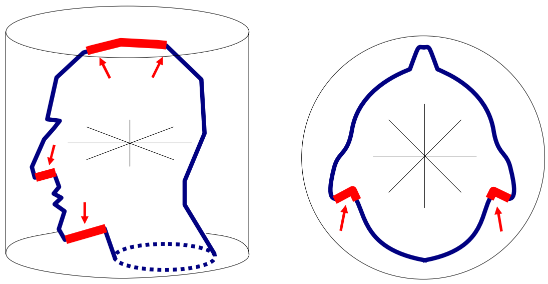
Cylindrical Textures

Common approach:

- created from input photographs:
  - L. Williams: “*Performance-Driven Facial Animation*”, SIGGRAPH ’90, 235-242, Aug. 1990
  - F. Pighin et al.: “*Synthesizing Realistic Facial Expressions from Photographs*”, SIGGRAPH ’98, 75-84, July 1998
- acquired during range scanning process (→ Cyberware scanners)

Cylindrical Textures

A head is similar to a cylinder ...is it?



Cylindrical Textures

Problems:

- limited texture resolution (Cyberware)
- need accurate geometry for registration (from photos)
- visual artifacts:
  - on top of the head
  - behind the ears
  - under the chin
- limited animation (eyes, teeth)

Textures from Photographs

Given:

- 3D mesh
- uncalibrated images (digitized photographs)

Assumptions:

- mesh represents real object (head) sufficiently precise
- images cover all areas of real object

Solution:

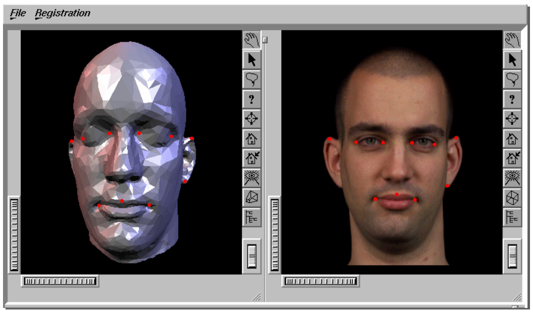
- C. Rocchini et al.: “*Multiple Textures Stitching and Blending on 3D Objects*”, EG Rendering Workshop ’99
  - register images, create texture patches

Tsai Algorithm

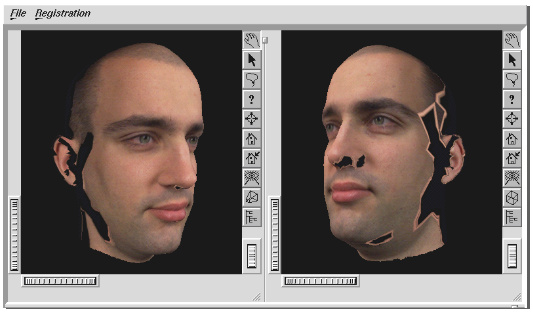
R. Y. Tsai: "*A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses*", IEEE J. of Robotics and Automation, RA-3(4), Aug. 1987

- compute intrinsic camera parameters (effective focal length, radial distortion, optical center) *once* from images of calibration pattern for different points of view using non-linear optimization
- compute extrinsic camera parameters (rotation & translation) *for each input image* using corresponding points (3D geometry ⇔ 2D image) and linear optimization

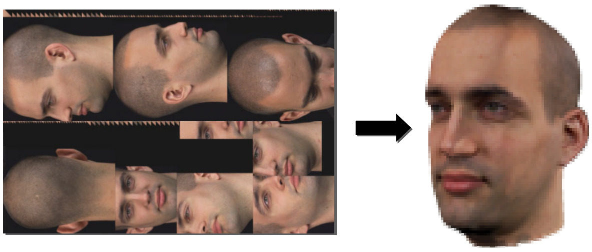
Corresponding Points



Texture Binding



Texture Combination



Important aspects:

- optimal packing of individual segments
- smooth transition between segments (*blending*)

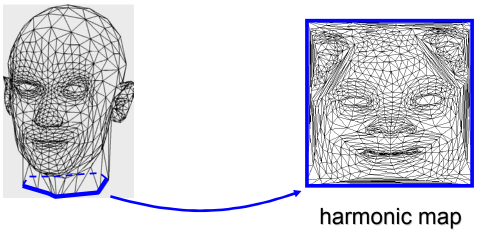
Texture Atlases

Problems:

- not suitable for mip-mapping
- waste of texture space:
  - optimal packing of patches is difficult
  - patches contain redundant information

A Different Parameterization

A head is topologically similar to a disk:



Harmonic Maps

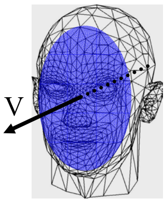
Characteristics:

- results in single texture patch suitable for mip-mapping
- 3D object must be topologically equivalent to a disk
- need to control distortion, e.g.:
  - P. V. Sander et al.: "*Texture Mapping Progressive Meshes*", SIGGRAPH '01, 409-416, Aug. 2001
- may introduce additional weights

Weighted Parameterization

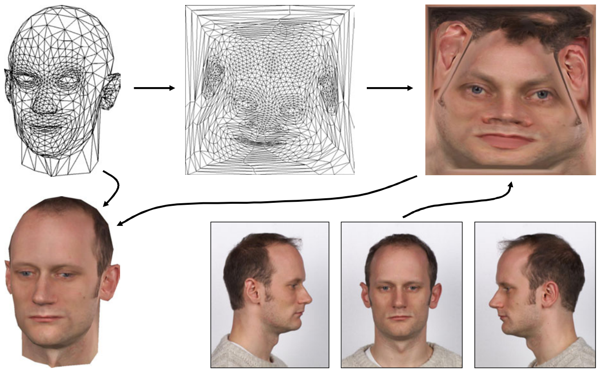
Facial region is most important:

- assign amount of texture space through weights



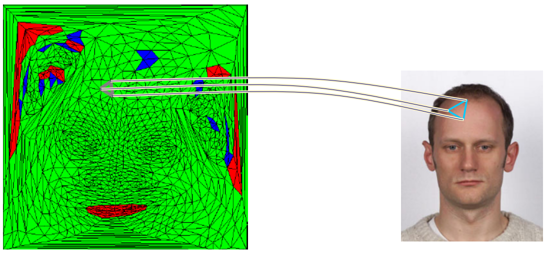
- triangles on the face become larger in the texture, backfacing triangles become smaller
- weights are computed automatically using dot product of triangle normal and viewing direction V of head model

Process Overview



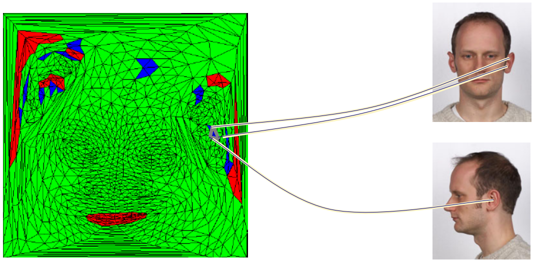
Texture Resampling: Resampling

common image for all vertices: resample triangle



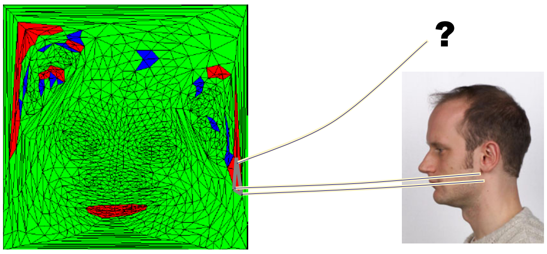
Texture Resampling: Interpolation

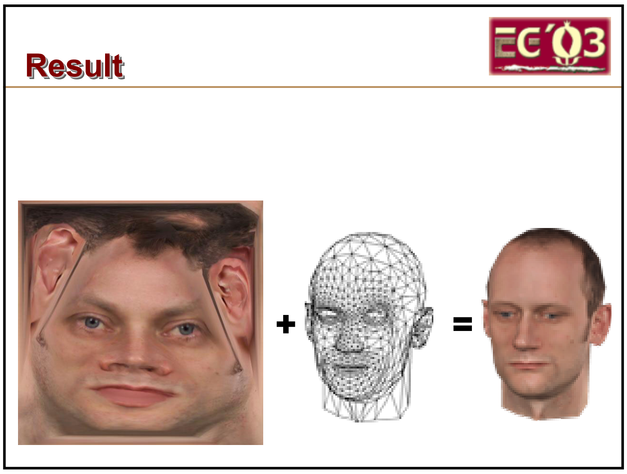
all vertices bound, no common image: interpolate



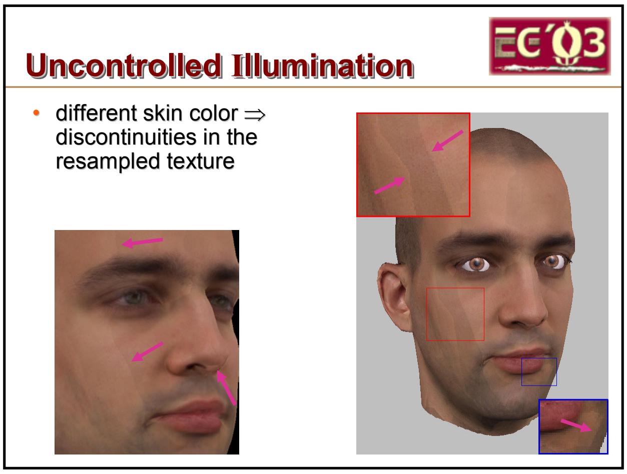
Texture Resampling: Filling holes

unbound vertices: apply iterative interpolation scheme



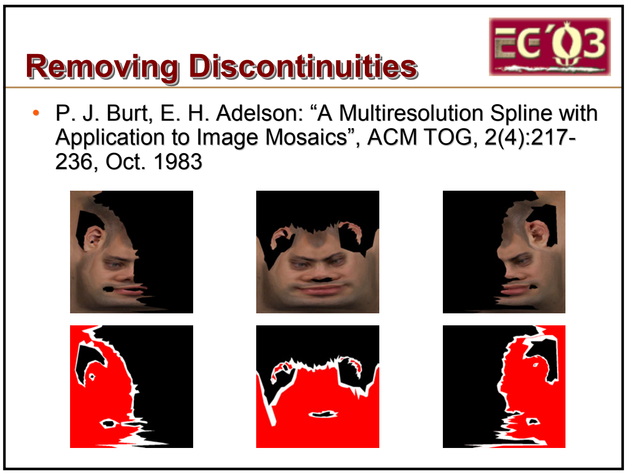


Result



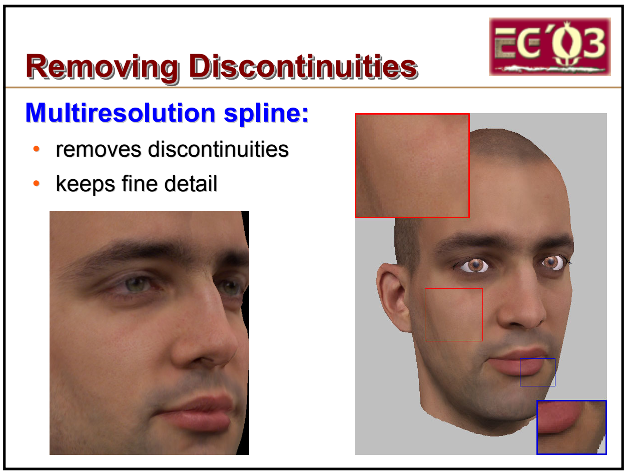
Uncontrolled Illumination

- different skin color ⇒ discontinuities in the resampled texture



Removing Discontinuities

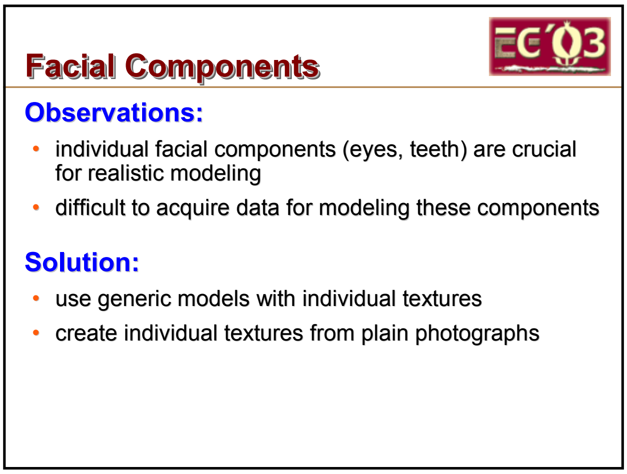
- P. J. Burt, E. H. Adelson: “A Multiresolution Spline with Application to Image Mosaics”, ACM TOG, 2(4):217-236, Oct. 1983



Removing Discontinuities

Multiresolution spline:

- removes discontinuities
- keeps fine detail



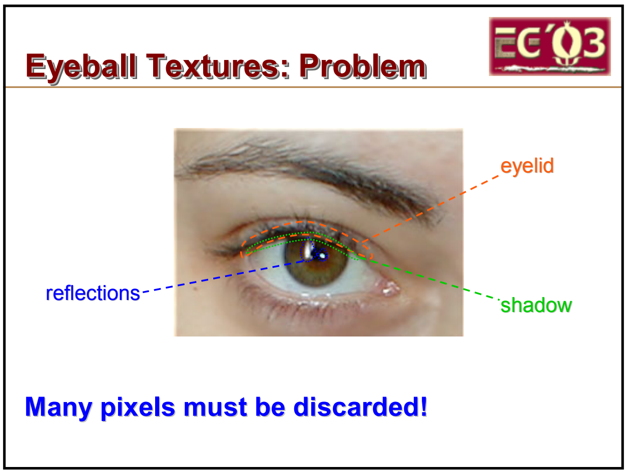
Facial Components

Observations:

- individual facial components (eyes, teeth) are crucial for realistic modeling
- difficult to acquire data for modeling these components

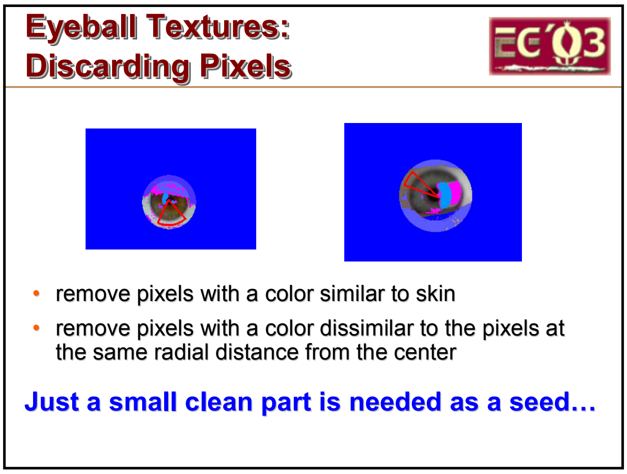
Solution:

- use generic models with individual textures
- create individual textures from plain photographs



Eyeball Textures: Problem

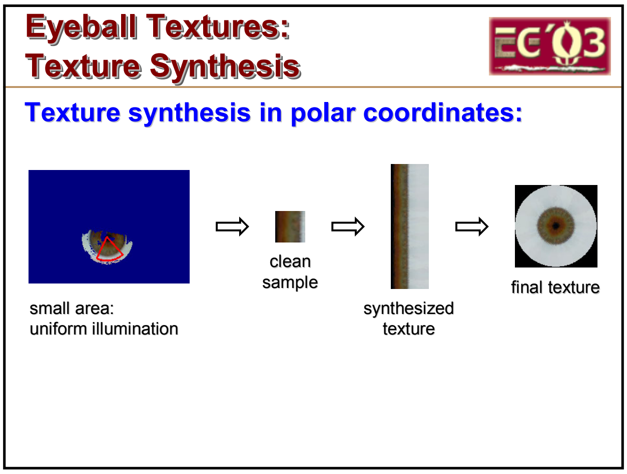
Many pixels must be discarded!



Eyeball Textures: Discarding Pixels

- remove pixels with a color similar to skin
- remove pixels with a color dissimilar to the pixels at the same radial distance from the center

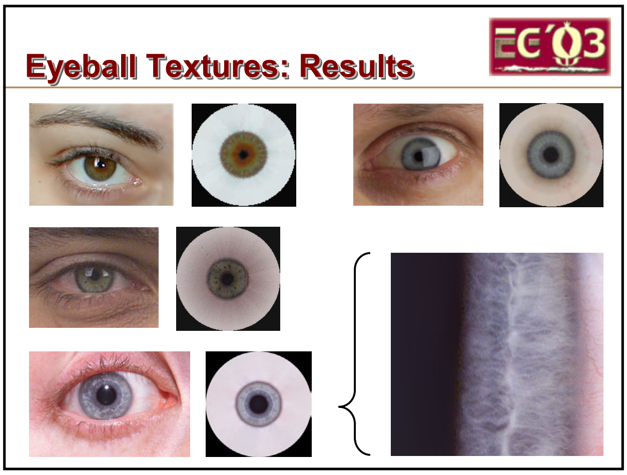
Just a small clean part is needed as a seed…



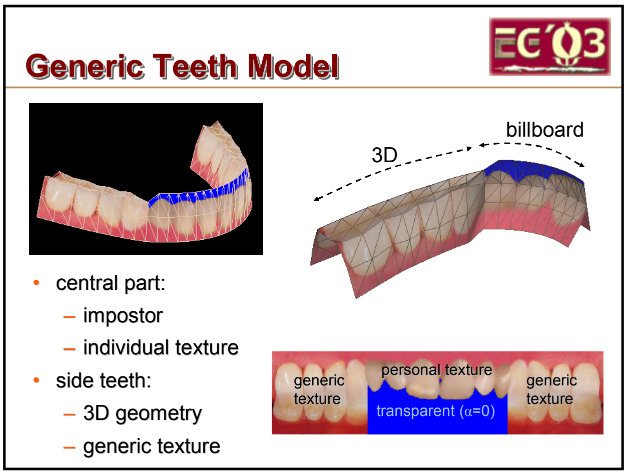
Eyeball Textures: Texture Synthesis

Texture synthesis in polar coordinates:

small area: uniform illumination ⇨ clean sample ⇨ synthesized texture ⇨ final texture



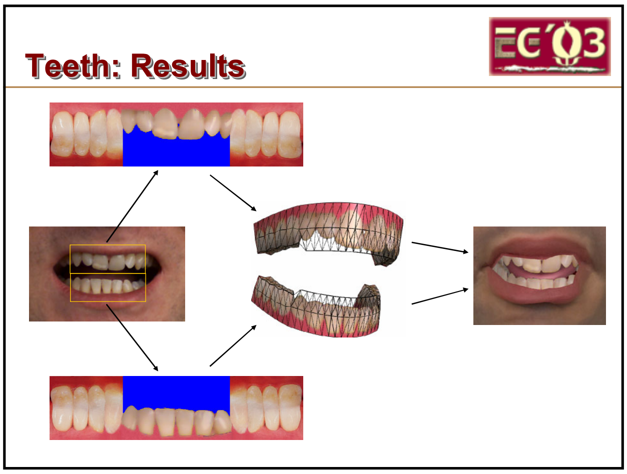
Eyeball Textures: Results



Generic Teeth Model

3D
billboard
generic texture
personal texture
transparent (α=0)
generic texture

- central part:
  - impostor
  - individual texture
- side teeth:
  - 3D geometry
  - generic texture



Teeth: Results

Texturing Faces

Marco Tarini[1,2] Hitoshi Yamauchi[1] Jörg Haber[1] Hans-Peter Seidel[1]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[2] Visual Computing Group, IEI, CNR Pisa, Italy
mtarini@di.unipi.it, {hitoshi,haberj,hpseidel}@mpi-sb.mpg.de

Abstract

We present a number of techniques to facilitate the generation of textures for facial modeling. In particular, we address the generation of facial skin textures from uncalibrated input photographs as well as the creation of individual textures for facial components such as eyes or teeth. Apart from an initial feature point selection for the skin texturing, all our methods work fully automatically without any user interaction. The resulting textures show a high quality and are suitable for both photo-realistic and real-time facial animation.

Key words: texture mapping, texture synthesis, mesh parameterization, facial modeling, real-time rendering

1 Introduction

Over the past decades, facial modeling and animation has achieved a degree of realism close to photo-realism. Although the trained viewer is still able to detect minor flaws in both animation and rendering of recent full-feature movies such as *Final Fantasy*, the overall quality and especially the modeling and texturing are quite impressive. However, several man-years went into the modeling of each individual character from that movie. Trying to model a real person becomes even more tricky: the artistic licence to create geometry and textures that "look good" is replaced by the demand to create models that "look real".

A common approach towards creating models of real persons for facial animation uses range scanners such as, for instance, Cyberware scanners to acquire both the head geometry and texture. Unfortunately, the texture resolution of such range scanning devices is often low compared to the resolution of digital cameras. In addition, the textures are typically created using a cylindrical projection. Such cylindrical textures have the drawback to introduce visual artifacts, for instance on top of the head, behind the ears, or under the chin. Finally, there is no automatic mechanism provided to generate textures for individual facial components such as eyes and teeth.

In this paper, we present an approach to generate high-resolution textures for both facial skin and facial components from several uncalibrated photographs. The generation of these textures is automated to a large extent, and the resulting textures do not exhibit any patch structures, i.e. they can be used for mip-mapping. Our approach combines several standard techniques from texture mapping and texture synthesis. In addition, we introduce the following contributions:

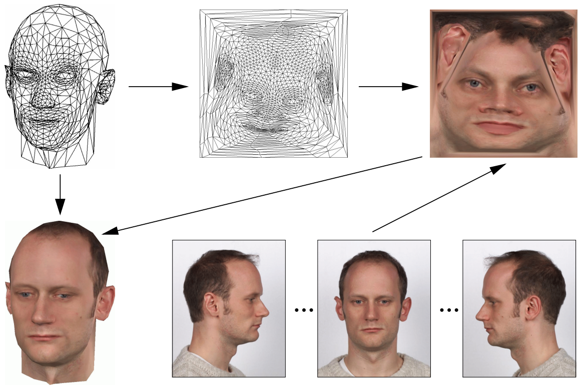


Figure 1: Overview of our skin texture generation process: the 3D face mesh is parameterized over a 2D domain and the texture is resampled from several input photographs.

- a view-dependent parameterization of the 2D texture domain to enhance the visual quality of textures with a fixed resolution;
- a texture resampling method that includes color interpolation for non-textured regions and visual boundary removal using multiresolution splines with a fully automatic mask generation;
- a radial texture synthesis approach with automatic center finding, which robustly produces individual eyeball textures from a single input photograph;
- a technique that uses a single natural teeth photograph to generate a teeth texture, which is applied to an appropriate 3D model to resemble the appearance of the subject's mouth.

appeared originally in Proc. Graphics Interface 2002, pp. 89–98

All of these techniques are fully automated to minimize the construction time for creating textures for facial modeling. However, we do not address the topic of facial modeling itself in this paper. We apply the textures generated by the techniques presented in this paper in our facial animation system [12], which has been designed to produce physically based facial animations that perform in real-time on common PC hardware. Thus the focus of our texture generation methods is primarily on the applicability of the textures for OpenGL rendering and a simple but efficient acquisition step, which does not require sophisticated camera setups and calibration steps.

2 Previous and Related Work

Research on either texturing or facial animation has provided a large number of techniques and insights over the years, see the surveys and textbooks in [13, 6] and [25] for an overview. Texturing in the context of facial animation is, however, an often neglected issue. Many sophisticated facial animation approaches, e.g. [32, 18, 19], simply use the textures generated by Cyberware scanners. In [35], Williams presents an approach to generate and register a cylindrical texture map from a peripheral photograph. This approach is meanwhile superseded by the ability of Cyberware scanners to acquire geometry and texture in one step. The method presented in [1] generates an individual head geometry and texture by linear combination of head geometries and textures from a large database that has been acquired using a Cyberware scanner in a costly preprocessing step. Marschner *et al.* describe a technique that uses several input photographs taken under controlled illumination with known camera and light source locations to generate an albedo texture map of the human face along with the parameters of a BRDF [23]. Several other approaches such as [26, 11, 16, 17] are image-based and use a small number of input photographs (or video streams) for the reconstruction of both geometry and texture. Although these approaches could potentially yield a higher texture quality compared to the Cyberware textures, they typically suffer from a less accurate geometry reconstruction, limited animation, and reduced texture quality by using cylindrical texture mapping.

Creating textures from multiple, unregistered photographs has been addressed in the literature by several authors [28, 3, 24]. First, they perform a camera calibration for each input photograph based on corresponding feature points. Next, a texture patch is created for each triangle of the input mesh. The approaches differ in the way these texture patches are created, blended, and combined into a common texture. However, the resulting textures always exhibit some patch structure, which makes it impossible to generate mip-maps from these textures. Creating textures that can be mip-mapped requires to construct a parameterization of the mesh over a two-dimensional domain. To this end, generic techniques based on spring meshes have been presented in [10, 15, 7]. Special parameterizations that minimize distortion during texture mapping for different kinds of surfaces have been investigated by several authors, see for instance [27, 29, 22, 21].

Texture synthesis [9, 33] has become an active area of research in the last few years. Recent publications focus on texture synthesis on surfaces [34, 31, 36] or on texture transfer [8, 14]. All of the methods presented so far use a Euclidean coordinate system for the synthesis of textures. In contrast, we use a polar coordinate system to synthesize textures that exhibit some kind of radial similarity.

3 Texturing Facial Skin

To generate a skin texture for a head model, we first take about three to five photographs of the person's head from different, uncalibrated camera positions. All photographs are taken with a high-resolution digital camera (3040×2008 pixels). The camera positions should be chosen in such a way that the resulting images roughly cover the whole head. During the acquisition, no special illumination is necessary. However, the quality of the final texture will benefit from a uniform, diffuse illumination. In addition, we acquire the geometry of the head using a structured-light range scanner. As a result, we obtain a triangle mesh that consists of up to a few hundred thousand triangles. After the texture registration step, this triangle mesh is reduced to about 1.5k triangles for real-time rendering using a standard mesh simplification technique. Each photograph is registered with the high-resolution triangle mesh using the camera calibration technique developed by Tsai [30]. Since the intrinsic parameters of our camera/lens have been determined with sub-pixel accuracy in a preprocessing step, we need to identify about 12–15 corresponding feature points on the mesh and in the image to robustly compute the extrinsic camera parameters for each image. This manual selection of feature points is the only step during our texture generation process that requires user interaction.

Next, we automatically construct a parameterization of the 3D input mesh over the unit square $[0, 1]^2$. This step is described in detail in the following Section 3.1. Finally, every triangle of the 2D texture mesh is resampled from the input photographs. A multiresolution spline method is employed to remove visual boundaries that might arise from uncontrolled illumination conditions during the photo session. Details about this resampling and blending step are given in Section 3.2. Figure 1 shows an overview of our texture generation process.

3.1 Mesh Parameterization

We want to parameterize the 3D input mesh over the 2D domain $[0,1]^2$ in order to obtain a single texture map for the whole mesh. To obtain a mip-mappable texture, the texture should not contain individual patches (*texture atlas*) but rather consist of a single patch. Clearly, this goal cannot be achieved for arbitrary meshes. In our case, the face mesh is topologically equivalent to a part of a plane, since is has a boundary around the neck and does not contain any handles. Thus we can “flatten” the face mesh to a part of a plane that is bounded by its boundary curve around the neck. We represent the original face mesh by a spring mesh and use the L^2 stretch norm presented in [29] to minimize texture stretch. In our simulations, this L^2 norm performs better than the L^∞ norm that is recommended by the authors of [29].

By applying the texture stretch norm, texture stretch is minimized over the whole mesh. In the following step, we introduce some controlled texture stretch again. Since the size of textures that can be handled by graphics hardware is typically limited, we would like to use as much texture space as possible for the “important” regions of a head model while minimizing the texture space allocated to “unimportant” regions. Obviously, the face is more important for the viewer than the ears or even the back of the head. To accomplish some biased texture stretch, we have introduced an additional weighting function ω into the L^2 stretch norm presented in [29]:

$$L^2(M) \ := \ \sqrt{\frac{\sum\limits_{T_i \in M} (L^2(T_i))^2\, \omega(T_i) A'(T_i)}{\sum\limits_{T_i \in M} \omega(T_i) A'(T_i)}}$$

with

$$\omega(T_i) \ := \ \frac{1}{\langle N(T_i), V \rangle + k} \, ,$$

where $M = \{T_i\}$ denotes the triangle mesh, $A'(T_i)$ is the surface area of triangle T_i in 3D, $N(T_i)$ is the triangle normal of T_i, V is the direction into which the head model looks, and $k > 1$ is a weighting parameter. The weighting function ω thus favors the triangles on the face by diminishing their error while penalizing the triangles on the back of the head by amplifying their error. As a consequence, triangles on the face become larger in the texture mesh while backfacing triangles become smaller. Useful values for k are from within $[1.01, 2]$.

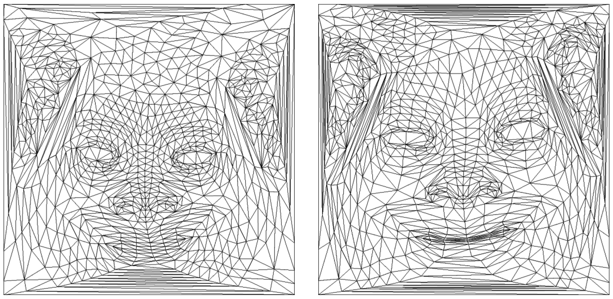


Figure 2: Comparison between a view-independent texture mesh parameterization according to [29] (left) and our view-dependent parameterization (right).

Figure 2 shows a view-independent texture mesh parameterization obtained with the original L^2 stretch norm as well as a view-dependent parameterization with our modified stretch norm for $k = 1.2$.

The difference between our *view-dependent texture mesh parameterization* and the *view-dependent texture mapping* proposed in [5, 26] is the following: the latter performs an adaptive blending of several photographs for each novel view, whereas we create a static texture that has its texture space adaptively allocated to regions of different visual importance.

3.2 Texture Resampling

After having created the 2D texture mesh from the 3D face mesh, we resample the texture mesh from the input photographs that have been registered with the face mesh. First, we perform a vertex-to-image binding for all vertices of the 3D face mesh. This step is carried out as suggested in [28]: Each mesh vertex v is assigned a set of *valid photographs*, which is defined as that subset of the input photographs such that v is visible in each photograph and v is a non-silhouette vertex. A vertex v is visible in a photograph, if the projection of v on the image plane is contained in the photograph **and** the normal vector of v is directed towards the viewpoint **and** there are no other intersections of the face mesh with the line that connects v and the viewpoint. A vertex v is called a silhouette vertex, if at least one of the triangles in the fan around v is oriented opposite to the viewpoint. For further details see [28]. In contrast to the approach in [28], we do not require that all vertices of the face mesh are actually bound to at least one photograph, i.e. the set of valid photographs for a vertex may be empty.

Let $\triangle = \{v_1, v_2, v_3\}$ denote a triangle of the face mesh and $\widetilde{\triangle} = \{\tilde{v}_1, \tilde{v}_2, \tilde{v}_3\}$ be the corresponding triangle in the texture mesh. For each triangle $\triangle$, exactly one of the following situations might occur (see also Figure 3):

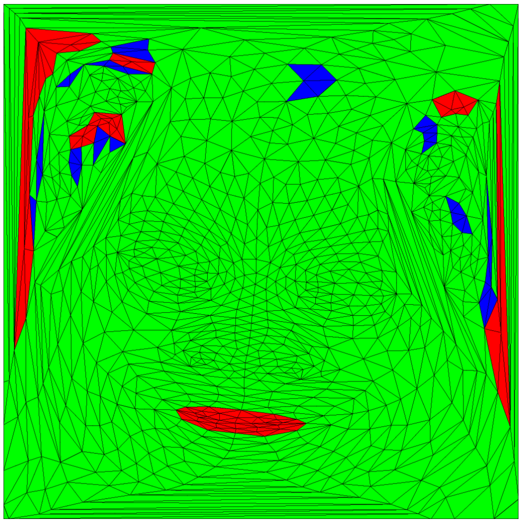


Figure 3: Color-coded triangles of the texture mesh: each green triangle has at least one common photograph to which all of its vertices are bound; the vertices of blue triangles don't have a common photograph, but they are all bound; red triangles have at least one unbound vertex.

1. There exists at least one common photograph in the sets of valid photographs of the three vertices v_1, v_2, v_3 of $\triangle$ (green triangles).

2. All of the vertices of $\triangle$ are bound to at least one photograph, but no common photograph can be found for all three vertices (blue triangles).

3. At least one vertex of $\triangle$ is not bound to any photograph (red triangles).

In the first case, we rasterize $\widetilde{\triangle}$ in texture space. For each texel T, we determine its barycentric coordinates ρ, σ, τ w.r.t. $\widetilde{\triangle}$ and compute the corresponding normal N by interpolating the vertex normals of $\triangle$: $N = \rho N(v_1) + \sigma N(v_2) + \tau N(v_3)$. For each common photograph i in the sets of valid photographs of all vertices of $\triangle$, we compute the dot product between N and the viewing direction V_i for the pixel P_i that corresponds to T. Finally, we color T with the color obtained by the weighted sum of pixel colors $\sum_i \langle N, V_i \rangle \cdot \text{Color}(P_i) \;/\; \sum_i \langle N, V_i \rangle$.

In the second case, we color each vertex $\tilde{v}_j$ of $\widetilde{\triangle}$ individually by summing up the weighted pixel colors of the corresponding pixels in all valid photographs i of $\tilde{v}_j$ similarly as in the first case: $\text{Color}(\tilde{v}_j) := \sum_i \langle N(v_j), V_i \rangle \cdot \text{Color}(P_i) \;/\; \sum_i \langle N(v_j), V_i \rangle$. The texels of the rasterization of $\widetilde{\triangle}$ are then colored by barycentric interpolation of the colors of the vertices $\tilde{v}_1, \tilde{v}_2, \tilde{v}_3$. Alternatively, we tried to use as much information as possible from the input photographs if, for instance, the vertices v_1, v_2 of $\triangle$ share a photograph and the vertices v_2, v_3 share another photograph. However, we found that this second case does not occur very often (cf. Figure 3) and that the difference between plain color interpolation and a more sophisticated approach is almost invisible.

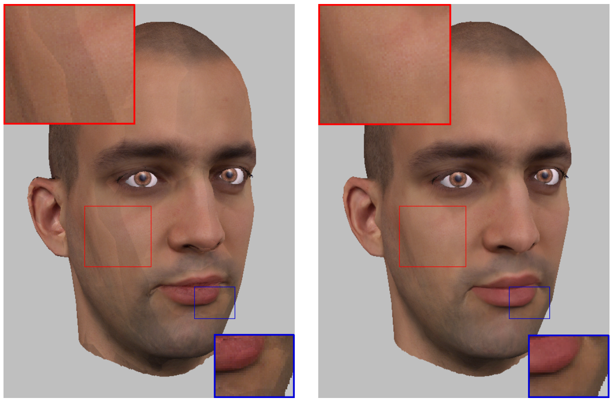


Figure 4: Boundaries in the skin texture (left) are removed using multiresolution spline techniques (right).

Since we do not require that each vertex of the face mesh is bound to at least one photograph, there might exist some vertices that cannot be colored by any of the previously described schemes. We address this problem in a two-stage process: First, we iteratively assign an interpolated color to each unbound vertex. Next, we perform the color interpolation scheme from the second case for the remaining triangles of $\widetilde{\triangle}$ that have not yet been colored. The first step iteratively loops over all unbound and uncolored vertices of the face mesh. For each unbound vertex v, we check if at least $p = 80\,\%$ of the vertices in the one-ring around v are colored (either by being bound to a photograph or by having an interpolated color). If this is true, we assign to v the average color of all the colored vertices around v, otherwise we continue with the next unbound vertex. We repeat this procedure until there are no further vertex updates. Next, we start the same procedure again, but this time we only require $p = 60\,\%$ of the vertices in the one-ring around v to be colored. As soon as there are no more updates, we repeat this step twice again with $p = 40\,\%$ and $p = 20\,\%$. Finally, we update each unbound vertex that has at least one colored neighbor. Upon termination of this last step, all vertices of the face mesh are either bound or colored and the remaining triangles of $\widetilde{\triangle}$ can be colored.

If the input photographs have been taken under uncontrolled illumination, the skin color might differ noticeably between the images. In this case, boundaries might appear in the resampled texture. We then apply a multires-

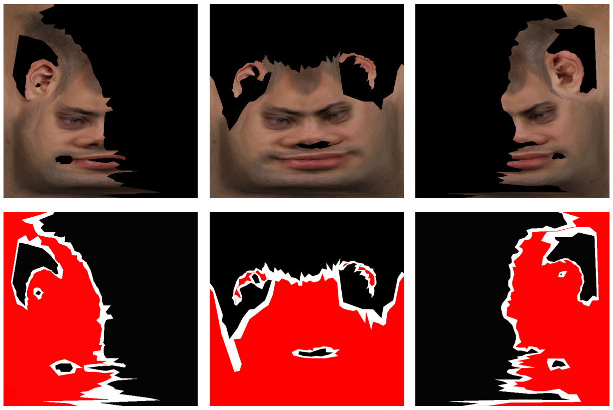


Figure 5: Multiresolution spline masks: three different regions in the texture mesh resampled from different input photographs (top) and their corresponding masks shown in red (bottom).

olution spline method as proposed in [2, 17] to remove visual boundaries. Figure 4 shows a comparison between a textured head model with and without multiresolution spline method applied. To smoothly combine texture regions that have been resampled from different input photographs, we automatically compute a mask for each region by removing the outmost ring of triangles around the region, see Figure 5. Such a shrinking is necessary to ensure that there is still some valid color information on the outside of the mask boundary, because these adjacent pixels might contribute to the color of the boundary pixels during the construction of Gaussian and Laplacian pyramids. In addition to the masks for each input photograph, we create one more mask that is defined as the complement of the sum of all the other masks. This mask is used together with the resampled texture to provide some color information in those regions that are not covered by any input photograph (e.g. the inner part of the lips). As described above, these regions have been filled by color interpolation in the resampled texture. By blending all of the masked input photographs and the masked resampled texture with a multiresolution spline, we obtain a final texture with no visual boundaries and crispy detail.

4 Texturing Facial Components

Both human eyes and teeth are important for realistic facial animation while, at the same time, it is difficult to acquire data from a human being to precisely model these facial components. Thus we use generic models of these components as shown in Figure 8. The design of our generic models has been chosen such that they look convincingly realistic when inserted into a face mesh while still being rendered efficiently using OpenGL hardware.

On the other hand, both eyes and teeth (especially the more visible middle ones) are crucial features to visually differentiate one individual from another. Hence, it would be very desirable to use individual models for each person. Luckily, texturing can do the trick alone: indeed it is sufficient to apply a personal texture to a generic model to get the desired effect. Moreover, it is possible to automatically and quickly generate these textures each from a single input photograph of the subject's eye and teeth, respectively. Details about this process will be given in the next two subsections.

4.1 Texturing Eyes

In order to realistically animate our head model, we must be able to perform rotations of the eyeball and dilation of the pupil. While the latter can be achieved by transforming the texture coordinates, we need an eye texture that covers the whole frontal hemisphere of the eyeball for the rotations.

Our goal to generate such an eyeball texture from a single input photograph is complicated by several factors such as the presence of occluding eyelids, shadows of eyelashes, highlights, etc. Still, all these factors are local and can be detected and removed. A new texture can then be synthesized from an input image consisting of the surviving pixels. In our current approach, we focus our effort on the iris, since it is obviously the most characteristic part of the eye.

Both the detection and the synthesis phase rely on the simplicity of the eye structure, i.e. an almost perfect point symmetry about the center, assuming our photograph represents an eye looking at the camera. To take advantage of this symmetry, we must first know precisely where the center of the eye is located. Since this would encumber the user, the center finding is done automatically by refining a rough estimation to sub-pixel precision using the following heuristic: we progressively enlarge an initially point-sized circle while checking the pixels on the circle at every iteration. If these pixels are too bright, they are assumed to be outside the iris and we thus move the center of the circle away from them. When most of the circle is composed by too bright pixels, we assume its center is the eye center and its radius is the iris radius. This approach runs robustly as long as the initial estimation is inside the pupil or the iris.

At this point, removal of occluded, shadowed, and highlighted pixels is done by:

- removing pixels with a color too similar to the skin;
- removing pixels with a color too dissimilar to the pixels at the same radial distance from the center.

For the second case, we compute the average color and standard deviation of the pixels at the same radial dis-

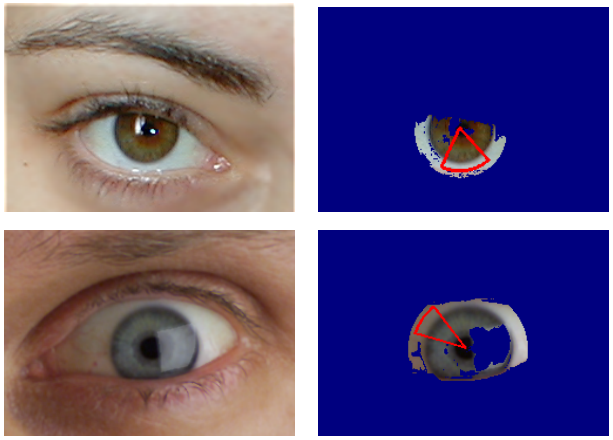


Figure 6: Two input photographs (left) and the resulting reference patches outlined by red sectors (right). Occluded, shadowed, highlighted, and skin-colored pixels (shown in blue) have been removed automatically.

tance and remove those pixels that are at least α times the standard deviation away from the average. The parameter α should be chosen within $[2, 3]$. We typically use a rather small value of $\alpha = 2.3$, as it empirically proved to remove the problematic (occluded, shadowed, highlighted, etc.) pixels in most cases. In addition, we remove pixels too close to the skin to better take into account small shadows cast by eyelids. Actually, the decision of which pixel to remove does not need excessively fine tuning: due to the regularity of the eye, we can be pretty conservative and remove many pixels, since the reconstruction phase requires only a small zone of pixels in order to synthesize more. Figure 6 shows the remaining set of pixels for two different input photographs.

For the reconstruction phase it is natural to resort to some texture synthesis from samples approach like e.g. [33]. In our case, we need to work in polar coordinates, because the eyeball texture behaves like a texture as defined in [33] only along the *angle* axis. This means that subregions of the eyeball texture are perceived to be similar if their *radius* coordinates are the same, cf. Figure 10. To take this into account, when choosing a candidate pixel p in the input image for filling a pixel p' in the output texture, we constrain the radius coordinate of p to be within a small threshold of the radius coordinate of p'.

A robust approach for texture synthesis is to use only a small patch of the original input image as the reference image and synthesize the texture from scratch. Although larger reference images theoretically result in more faithful textures, we obtained very good results with small reference patches covering a sector of about 30 degrees around the pupil. Small reference patches have the advantage of being more uniform and thus bypassing problems related to uneven lighting in the original photograph. In our approach, we simply use the largest sector of valid pixels of at most 60 degrees as the reference patch. In the rare cases where the largest sector is too small, e.g. spanning less than 20 degrees, the entire set of valid pixels with a valid neighborhood is used as the reference image.

Since the detail frequencies of human irises are roughly the same, it is sufficient to use a texture synthesis scheme with a fixed neighborhood size rather than a multiresolution approach. In our case, the size of the neighborhood mask depends only on the resolution of the input image. For instance, for an image of an iris with a diameter of approximately 80 pixels, we use a 3$\times$6 pixel mask (radius $\times$ angle). For other iris diameters, the pixel mask is set proportionally. Depending on the value of the radius coordinate, a neighborhood with a fixed size in polar coordinates covers areas of different sizes in the input image. Our simulations showed, however, that no correction is needed, since the human iris usually exhibits higher frequency detail towards the center. Thus an iris resampled in polar coordinates shows quite uniform frequency distribution. Figure 9 shows several input photographs together with the resulting eye textures for various individuals.

To speed-up the reconstruction step, we use a one-dimensional texture synthesis approach along the angle axis alone, modeling the texture as a Markov chain rather than a Markov random field. Each symbol of the chain is an entire row of texels at a given angle coordinate. We output each new row accordingly to the previous rows. This approach gives similar results (even if it requires slightly larger reference textures) and is much faster, not even requiring any vector quantization for finding the best neighborhood row. If, however, the size of the reference patch is very small, we apply a two-dimensional texture synthesis approach as described earlier in this section.

4.2 Texturing Teeth

Geometry and color of teeth are difficult to capture and, at the same time, crucial to reflect personal appearance. We address this problem by distinguishing between

- the six middle teeth (incisors and canines) and
- the rest of the teeth (4–5 on each side).

The middle teeth are much more visible than the other teeth. This means that they account for most of the visual appearance of an individual person, but also that it is much easier to reconstruct them from a photograph. In addition, the middle teeth have an almost two-dimensional structure: they are shaped to have the function of a blade. Their small width allows us to model

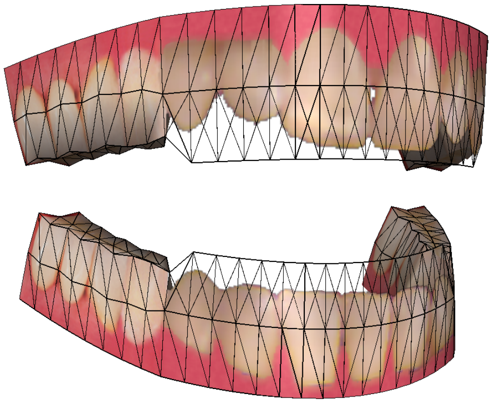


Figure 7: Teeth arch model using the texture shown in Figure 11. The wireframe shows the geometry of the teeth model, which consists of 384 triangles.

them using a billboard (impostor). Being a 2D data structure, the billboard can be easily extracted directly from a normal photograph of the subject exposing the teeth in a similar way as shown in Figure 11 (left). Using local transparency, it is straightforward to make the texture embed the teeth shape and size including gaps between teeth. This approach allows us to use the same (billboarded) 3D model for every face model and just change the texture from person to person.

The rest of the teeth, while being more voluminous and less accessible and visible, do not allow this useful shortcut. But, for the same reason, it is also less important to model them faithfully and individually for each single person. Thus it seems reasonable to use a standard 3D model and a standard texture (up to recoloring, see below) for this part of the teeth arch.

Following these considerations, we have built a generic 3D model for the teeth, which is non-uniformly scaled according to the individual skull and jaw geometry to fit into every head model. For each individual head model, we only need to vary the texture (including the billboard), which is created fully automatically. The generic teeth model is constructed such that the transition between the billboard (in the middle) and the 3D structure (left and right) is smooth, see Figure 7. The billboard, which is bent for better realism, could cause undesired artifacts when seen from above. To avoid this, only the upper part of the lower teeth and the lower part of the upper ones is actually modeled as a billboard. The remaining parts of the upper and lower middle teeth smoothly gain some width as they go up and down, respectively.

To automatically create a texture for the teeth, we start from a normal photograph of the subject showing his/her teeth. Several stages of the whole process of generating a teeth texture are shown in Figure 11. We color-code dark parts that represent voids with a blue color, which is replaced by a transparent alpha value during rendering. Similarly, we identify and remove gums, lips, and skin, recoloring it with some standard gums color. To make this color-coding more robust, we identify the different regions using threshold values, which are obtained by finding the biggest jumps in the histograms of the color distances to the target color (red for gums and black for voids). In addition, we expand teeth into those parts of the gums that have been covered by the lips in the input photograph. We use some simple heuristics to include the missing part of the tooth roots, cf. Figure 11.

During rendering, our teeth model is shaded using a Phong shading model, which means that we have to deshade our teeth texture. In order to do so for uncontrolled illumination, we equalize the color of the teeth, supposing they have approximately the same albedo. First, we define a target color by computing the average color of all teeth pixels and setting its brightness (but not the hue) to a predefined value. Next, we subdivide the texture in six vertical stripes and compute the average color of each stripe. We then add to the pixels in each column the difference between the target color and the stripe average, taking care of enforcing continuity in this correction by using a piecewise linear function. Similarly, we use the target color to correct the color of the "generic" part of the texture, which is applied to the side teeth. Finally, we composite the middle teeth texture into our generic texture using a curved boundary that follows the silhouettes of the canines.

5 Results

We have created facial textures for several individuals who have also been range-scanned to acquire their head geometry. Rendering of our head model is performed in real-time using OpenGL hardware (about 100 fps on a 1.7 GHz PC with a GeForce3 graphics board). A physics-based simulation is used to control the facial animation. Several images of our head models are distributed over this paper, see for instance Figures 1, 4, 8, and especially Figure 12. For each skin texture, the only interactive step is the initial identification of corresponding feature points. This step takes about five minutes per input photograph, which sums up to about 15–25 minutes spent interactively for three to five photographs. Computing an optimized parameterization of the face mesh (approx. 1600 triangles) takes about 80 minutes on a fast PC (1.7 GHz Pentium 4). Resampling a 2048×2048 texture from five input photographs takes about one minute, additional multiresolution spline blending (if necessary)

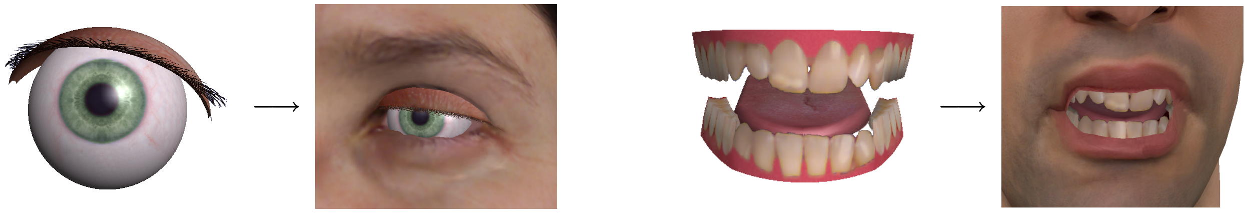


Figure 8: Generic models of eyes, teeth, and tongue are fitted into individual face meshes.

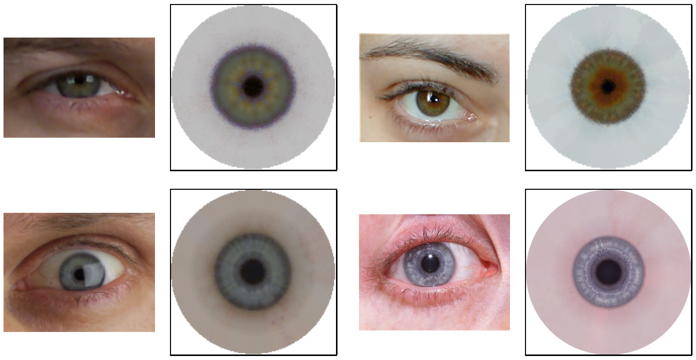


Figure 9: Input photographs and resulting eye textures: the input images have been taken under various illumination conditions with different resolutions. The size of the resulting textures changes from 128×128 (top left) to 1024×1024 (bottom right).

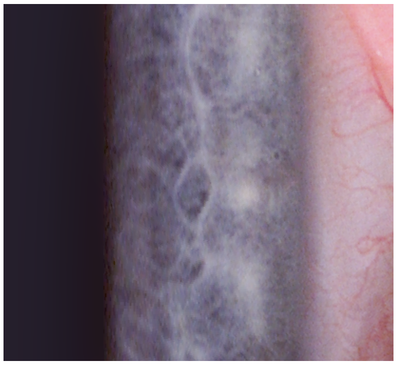


Figure 10: A detail of the texture from Figure 9 (bottom right) shown in polar coordinates. The abscissa represents the radius axis and the ordinate represents the angle axis.

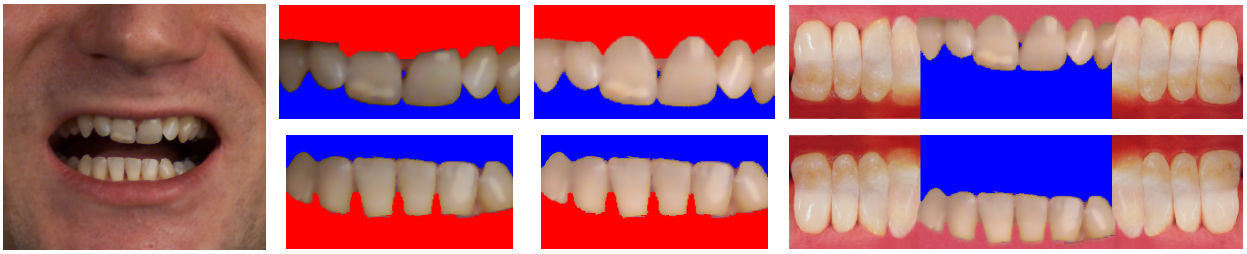


Figure 11: Teeth texture generation. Left to right: starting from an input photograph, we extract the upper and lower middle teeth, fill in missing parts and adjust the color, and composite the new image with a generic teeth texture. The blue pixels in the final texture (right) will be rendered transparently.

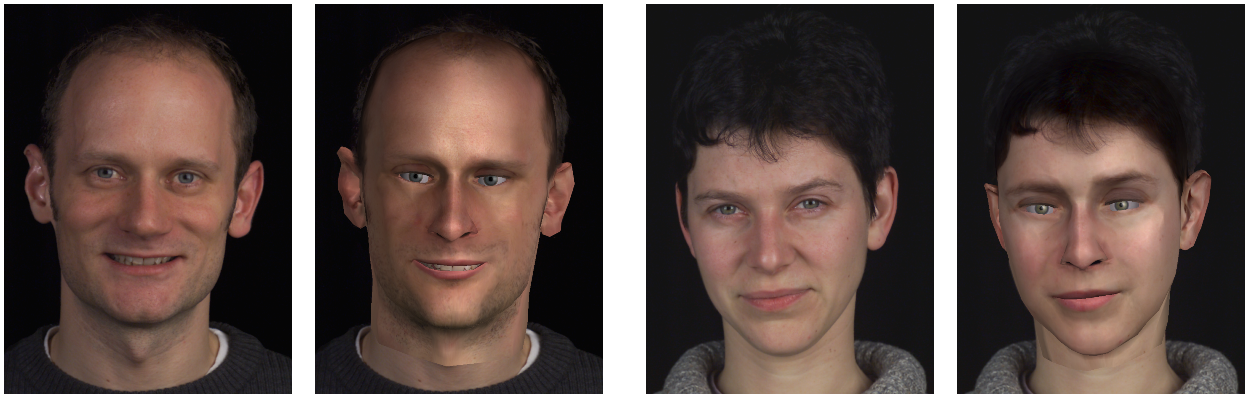


Figure 12: Side-by-side comparison of photographs (left) and head models (right) for plain OpenGL rendering.

takes about ten minutes. Currently, our algorithms are optimized with respect to robustness but not to speed.

Generating the teeth and eye textures takes only a few seconds even for large textures using the 1D Markov chain method for the texture synthesis. If a full Markov field is used, construction time may go up to several minutes, depending on the size of the texture being created.

6 Conclusion and Future Work

We have introduced a number of techniques that help to minimize the time and effort that goes into the creation of textures for facial modeling. With the exception of the initial feature point selection for the skin texturing, our methods are fully automated and do not require any user interaction.

For the generation of skin textures from uncalibrated input photographs, we propose a view-dependent parameterization of the texture domain and a texture resampling method including color interpolation for non-textured regions and multiresolution splining for the removal of visual boundaries. Using our methods, both eye and teeth textures can be created fully automatically from single input photographs, adding greatly to a realistic appearance of individual subjects during facial animation.

One of the main goals of ongoing research is to get rid of the interactive camera calibration step for skin texturing. Given that the resulting texture should contain fine detail, this is a tough problem, indeed. Automatic approaches such as [20] fail simply due to the fact that the silhouette of a human head looks more or less identical when viewed from within a cone of viewing directions from the front or the back. Furthermore, it would be desirable to account for lighting artifacts in the input photographs. Although a uniform, diffuse illumination during the photo session helps a lot, there are still contributions from diffuse and specular lighting in the photographs. Approaches to overcome these problems have been suggested [4, 23], but they require sophisticated camera setups and calibration steps. Finally, it would be very helpful to speed-up the computation time of the current bottleneck, namely the mesh parameterization, using a hierarchical coarse-to-fine approach.

Acknowledgments

The authors would like to thank their models Letizia, Claudia, and Kolja for all the smiles during the photo sessions. Many thanks also to our colleagues, who gave helpful comments during the development of our techniques, and to the anonymous reviewers for their suggestions.

References

[1] V. Blanz and T. Vetter. A Morphable Model for the Synthesis of 3D Faces. In *Computer Graphics (SIGGRAPH '99 Conf. Proc.)*, pages 187–194, August 1999.

[2] P. J. Burt and E. H. Adelson. A Multiresolution Spline with Application to Image Mosaics. *ACM Transactions on Graphics*, 2(4):217–236, October 1983.

[3] P. Cignoni, C. Montani, C. Rocchini, R. Scopigno, and M. Tarini. Preserving Attribute Values on Simplified Meshes by Resampling Detail Textures. *The Visual Computer*, 15(10):519–539, 1999.

[4] P. E. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Computer Graphics (SIGGRAPH '00 Conf. Proc.)*, pages 145–156, July 2000.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-based Approach. In *Computer Graphics (SIGGRAPH '96 Conf. Proc.)*, pages 11–20, August 1996.

[6] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley. *Texturing & Modeling: A Procedural Approach*. Academic Press, London, 2 edition, 1998.

[7] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. In *Computer Graphics (SIGGRAPH '95 Conf. Proc.)*, pages 173–182, August 1995.

[8] A. A. Efros and W. T. Freeman. Image Quilting for Texture Synthesis and Transfer. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, pages 341–346, August 2001.

[9] A. A. Efros and T. K. Leung. Texture Synthesis by Non-parametric Sampling. In *IEEE Int'l Conf. Computer Vision*, volume 2, pages 1033–1038, September 1999.

[10] M. S. Floater. Parametrization and Smooth Approximation of Surface Triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[11] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making Faces. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, pages 55–66, July 1998.

[12] J. Haber, K. Kähler, I. Albrecht, H. Yamauchi, and H.-P. Seidel. Face to Face: From Real Humans to Realistic Facial Animation. In *Proc. Israel-Korea Binational Conf. on Geometrical Modeling and Computer Graphics*, pages 73–82, October 2001.

[13] P. S. Heckbert. Survey of Texture Mapping. *IEEE Computer Graphics and Applications*, 6(11):56–67, November 1986.

[14] A. Hertzmann, Ch. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image Analogies. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, pages 327–340, August 2001.

[15] K. Hormann and G. Greiner. MIPS: An Efficient Global Parametrization Method. In *Curve and Surface Design: Saint-Malo 1999*, pages 153–162. Vanderbilt University Press, 2000.

[16] W.-S. Lee, J. Gu, and N. Magnenat-Thalmann. Generating Animatable 3D Virtual Humans from Photographs. In *Computer Graphics Forum (Proc. EG 2000)*, volume 19, pages C1–C10, August 2000.

[17] W.-S. Lee and N. Magnenat-Thalmann. Fast Head Modeling for Animation. *Image and Vision Computing*, 18(4):355–364, March 2000.

[18] Y. Lee, D. Terzopoulos, and K. Waters. Constructing Physics-based Facial Models of Individuals. In *Proc. Graphics Interface '93*, pages 1–8, May 1993.

[19] Y. Lee, D. Terzopoulos, and K. Waters. Realistic Modeling for Facial Animations. In *Computer Graphics (SIGGRAPH '95 Conf. Proc.)*, pages 55–62, August 1995.

[20] H. P. A. Lensch, W. Heidrich, and H.-P. Seidel. Automated Texture Registration and Stitching for Real World Models. In *Proc. Pacific Graphics 2000*, pages 317–326, October 2000.

[21] B. Lévy. Constrained Texture Mapping for Polygonal Meshes. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, pages 417–424, August 2001.

[22] J. Maillot, H. Yahia, and A. Verroust. Interactive Texture Mapping. In *Computer Graphics (SIGGRAPH '93 Conf. Proc.)*, pages 27–34, August 1993.

[23] S. R. Marschner, B. Guenter, and S. Raghupathy. Modeling and Rendering for Realistic Facial Animation. In *Rendering Techniques 2000 (Proc. 11th EG Workshop on Rendering)*, pages 231–242, 2000.

[24] P. J. Neugebauer and K. Klein. Texturing 3D Models of Real World Objects from Multiple Unregistered Photographic Views. In *Computer Graphics Forum (Proc. EG '99)*, volume 18, pages C245–C256, September 1999.

[25] F. I. Parke and K. Waters, editors. *Computer Facial Animation*. A K Peters, Wellesley, MA, 1996.

[26] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing Realistic Facial Expressions from Photographs. In *Computer Graphics (SIGGRAPH '98 Conf. Proc.)*, pages 75–84, July 1998.

[27] D. Piponi and G. D. Borshukov. Seamless Texture Mapping of Subdivision Surfaces by Model Pelting and Texture Blending. In *Computer Graphics (SIGGRAPH '00 Conf. Proc.)*, pages 471–478, July 2000.

[28] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. Multiple Textures Stitching and Blending on 3D Objects. In *Rendering Techniques '99 (Proc. 10th EG Workshop on Rendering)*, pages 119–130, 1999.

[29] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture Mapping Progressive Meshes. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, pages 409–416, August 2001.

[30] R. Y. Tsai. An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 364–374, June 1986.

[31] G. Turk. Texture Synthesis on Surfaces. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, pages 347–354, August 2001.

[32] K. Waters and D. Terzopoulos. Modeling and Animating Faces Using Scanned Data. *J. Visualization and Computer Animation*, 2(4):123–128, October–December 1991.

[33] L.-Y. Wei and M. Levoy. Fast Texture Synthesis Using Tree-Structured Vector Quantization. In *Computer Graphics (SIGGRAPH '00 Conf. Proc.)*, pages 479–488, July 2000.

[34] L.-Y. Wei and M. Levoy. Texture Synthesis over Arbitrary Manifold Surfaces. In *Computer Graphics (SIGGRAPH '01 Conf. Proc.)*, pages 355–360, August 2001.

[35] L. Williams. Performance-Driven Facial Animation. In *Computer Graphics (SIGGRAPH '90 Conf. Proc.)*, volume 24, pages 235–242, August 1990.

[36] L. Ying, A. Hertzmann, H. Biermann, and D. Zorin. Texture and Shape Synthesis on Surfaces. In *Rendering Techniques 2001 (Proc. 12th EG Workshop on Rendering)*, pages 301–312, 2001.

EUROGRAPHICS 2003

Advanced Rendering Techniques

Jörg Haber

Rendering Faces

- skin modeling and rendering:
  - bump mapping for skin dimples and wrinkles
  - shading: BRDFs, subsurface scattering
  - “delighting” textures: removing directional illumination components from textures
- hair modeling and rendering

Real-time or off-line rendering?

Skin Rendering

- different techniques for modeling/rendering skin:

✓ – simple geometry + texture

(✓) – simple geometry + bump mapping + texture

(✗) – simple geometry + BRDF

(✗) – simple geometry + bump mapping + BRDF

✗ – simple geometry + displacement mapping + texture

– complex geometry + texture/BRDF

✓ : suitable for graphics hardware

✗ : not suitable for graphics hardware

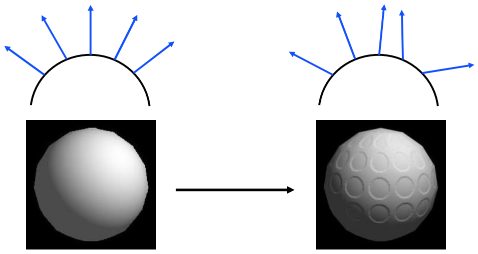
Lighting / Shading

- simulated using virtual light sources, e.g. point lights, directional light, spotlights
- two types of illumination models:
  - local models: color and intensity of a surface point are determined by local attributes (e.g. surface normal) and direct contribution of light sources
    ⇒ suitable for graphics hardware
  - global models: all scene objects participate in the illumination of all surfaces (shadows, mirroring, indirect illumination, color bleeding)
    ⇒ very expensive computation
- one of the most important parameters: surface normal

Bump Mapping

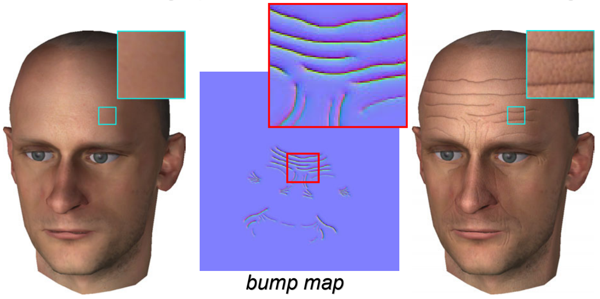
Idea:

- simulate complex geometry using coarse geometry and “faked” per-pixel surface normals



Rendering Wrinkles

- encode surface normals into RGB texture
- use modern graphics hardware for real-time rendering



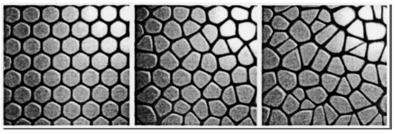
bump map

Rendering Skin

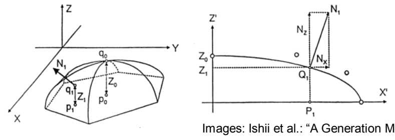
- T. Ishii et al.: “*A Generation Model for Human Skin Texture*”, Proc. CGI ‘93, 39-150, 1993
- presents a method for generating skin structure bump maps and an appropriate illumination model for rendering skin
- surface normals are computed from recursively generated, hierarchical micro-geometry during preprocessing
- illumination model simulates multi-layered skin structure taking into account subsurface scattering

“Pattern Generation”

- skin cells are represented by Voronoi cells



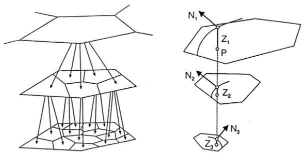
- every skin cell bulges upwards above its center; ridge shape: cubic Bézier curves

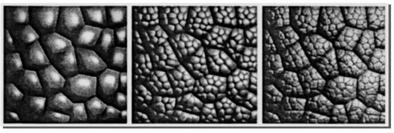


Images: Ishii et al.: “A Generation Model for Human Skin Texture”

“Hierarchical Skin Structure”

- recursive Voronoi subdivision of skin cells (3 levels)

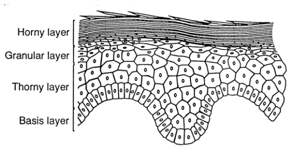


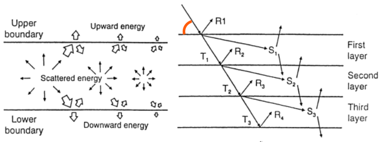


Images: Ishii et al.: “A Generation Model for Human Skin Texture”

“Multiple Light Reflections”

- multi-layered skin structure results in complex light transport mechanisms
- model: parallel layers; reflection & transmission & scattering at each layer boundary



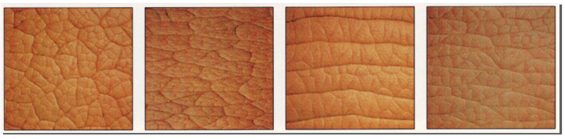


precompute lighting w.r.t. angle of incidence at skin surface

Images: Parke/Waters: “Computer Facial Animation” (1996)

Ishii et al.: Results

- generic model for rendering skin
- orientation of skin cells can be aligned to wrinkles
- anisotropic scaling of skin cells (→ wrist)
- skin structure can be rendered in real-time using graphics hardware bump mapping; illumination model not (yet) suitable for real-time rendering



Images: Ishii et al.: “A Generation Model for Human Skin Texture”

BRDF – What’s that?

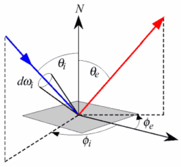
- BRDF = bidirectional reflectance distribution function, describes reflectance of a surface point as the ratio of radiance L and irradiance I :

$$f_r(\theta_i,\phi_i,\theta_e,\phi_e,\lambda)=\frac{\mathrm{d}L_\lambda(\theta_e,\phi_e)}{\mathrm{d}I_\lambda(\theta_i,\phi_i)}$$

- BRDF has 5 degrees of freedom:

$$f_r:\mathbf{R}^5\rightarrow\mathbf{R}$$

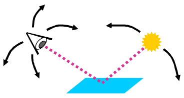
- can be simplified assuming isotropic materials and discrete spectrum (RGB instead of λ)

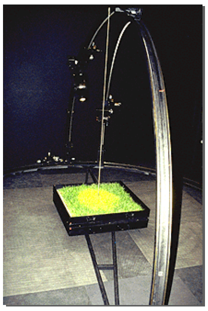


Measuring BRDFs (I)

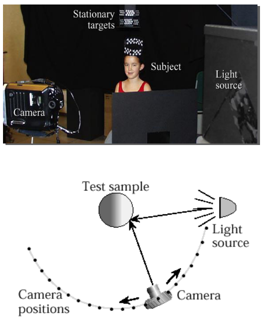
Gonioreflectometer:

- measures radiance for many different positions of material sample, light source, and sensor
- planar material sample
- 3 degrees of freedom:
  - position of light source (1D)
  - position of sensor (2D)





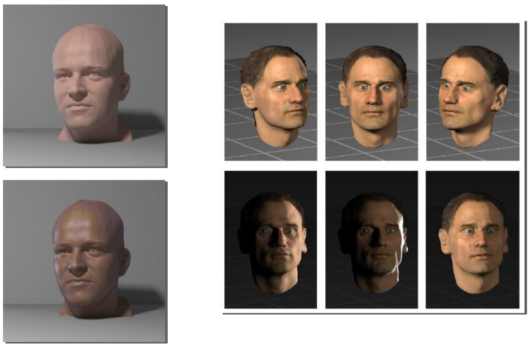
Measuring BRDFs (II)



- S. Marschner et al.: “Image-based BRDF Measurement Including Human Skin”, Proc. EG Rendering Workshop ‘99, 131-144, 1999.
- uses photographs taken under controlled illumination conditions
- needs geometry & normals from 3D range scanner
- “inverse ray tracing” computes BRDF for given positions of camera and light source and pixel intensity

Images: Marschner et al., “Image-based BRDF Measurement Including Human Skin”

Marschner et al.: Results

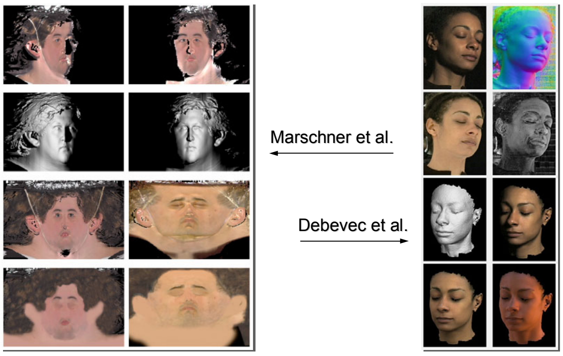


Images: Marschner et al., EG Rendering Workshop ’99 & 2000

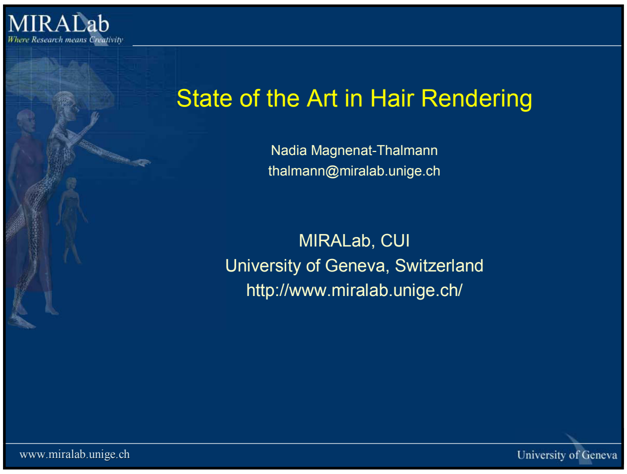
“De-lighting” Textures

- S. Marschner, B. Guenter, S. Raghupathy: *“Modeling and Rendering for Realistic Facial Animation”*, Proc. EG Rendering Workshop 2000, 231-242, June 2000
- P. Debevec et al.: *“Acquiring the Reflectance Field of a Human Head”*, SIGGRAPH 2000, 145-156, July 2000
- extract diffuse reflectivity (albedo map) from photographs
- photographs must be taken under controlled illumination conditions (relative position of object, camera, light sources)
- diffuse reflectivity is computed per texel from viewing direction, direction of incident light, surface normal and radiance (= color from photograph)

Results



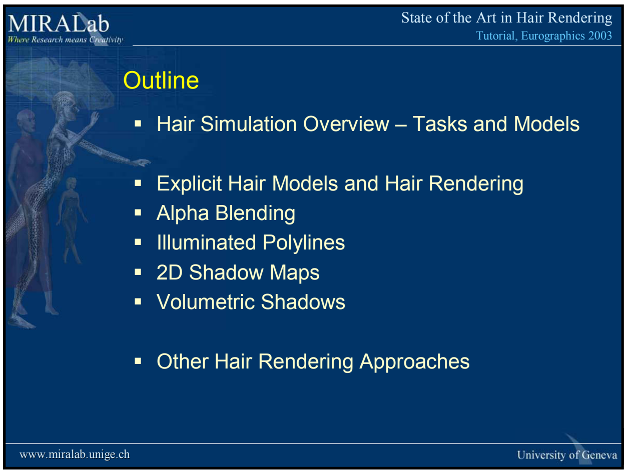
Images: Marschner et al., EG Rendering Workshop 2000 & Debevec et al., SIGGRAPH 2000



State of the Art in Hair Rendering

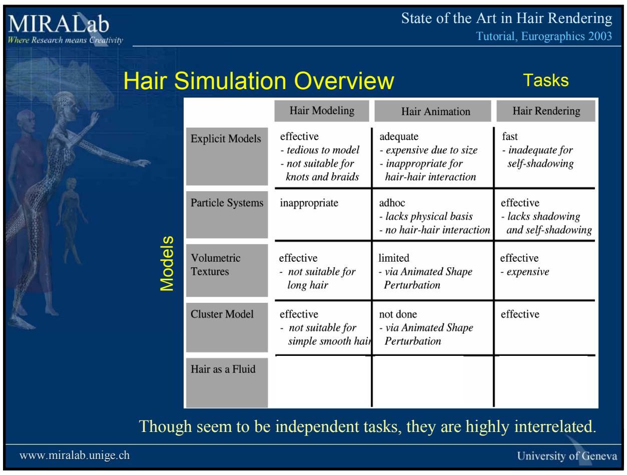
Nadia Magnenat-Thalmann
thalmann@miralab.unige.ch

MIRALab, CUI
University of Geneva, Switzerland
http://www.miralab.unige.ch/



Outline

- Hair Simulation Overview – Tasks and Models
- Explicit Hair Models and Hair Rendering
- Alpha Blending
- Illuminated Polylines
- 2D Shadow Maps
- Volumetric Shadows
- Other Hair Rendering Approaches



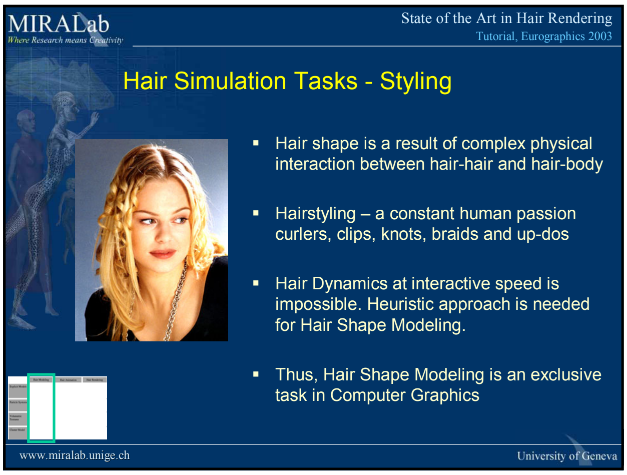
Hair Simulation Overview

Tasks

Models

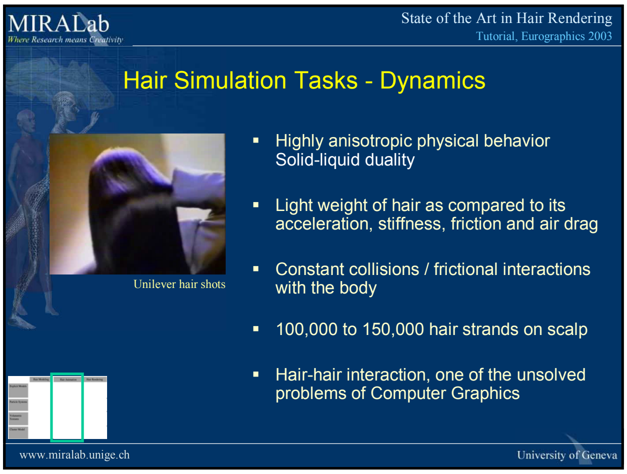
	Hair Modeling	Hair Animation	Hair Rendering
Explicit Models	effective *- tedious to model* *- not suitable for knots and braids*	adequate *- expensive due to size* *- inappropriate for hair-hair interaction*	fast *- inadequate for self-shadowing*
Particle Systems	inappropriate	adhoc *- lacks physical basis* *- no hair-hair interaction*	effective *- lacks shadowing and self-shadowing*
Volumetric Textures	effective *- not suitable for long hair*	limited *- via Animated Shape Perturbation*	effective *- expensive*
Cluster Model	effective *- not suitable for simple smooth hair*	not done *- via Animated Shape Perturbation*	effective
Hair as a Fluid			

Though seem to be independent tasks, they are highly interrelated.



Hair Simulation Tasks - Styling

- Hair shape is a result of complex physical interaction between hair-hair and hair-body
- Hairstyling – a constant human passion curlers, clips, knots, braids and up-dos
- Hair Dynamics at interactive speed is impossible. Heuristic approach is needed for Hair Shape Modeling.
- Thus, Hair Shape Modeling is an exclusive task in Computer Graphics



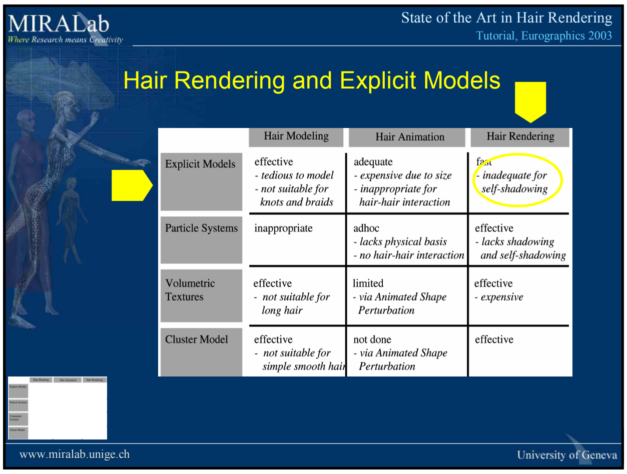
Hair Simulation Tasks - Dynamics

Unilever hair shots

- Highly anisotropic physical behavior Solid-liquid duality
- Light weight of hair as compared to its acceleration, stiffness, friction and air drag
- Constant collisions / frictional interactions with the body
- 100,000 to 150,000 hair strands on scalp
- Hair-hair interaction, one of the unsolved problems of Computer Graphics

Hair Simulation Tasks - Rendering

- Intricate geometry of individual hair
- Large number of hair strands
- Complex interaction with light and shadows multiple scattering, luster, self-shadowing
- Anisotropy in shading
- Small thickness of the hair – anti-aliasing How artists paint hair?



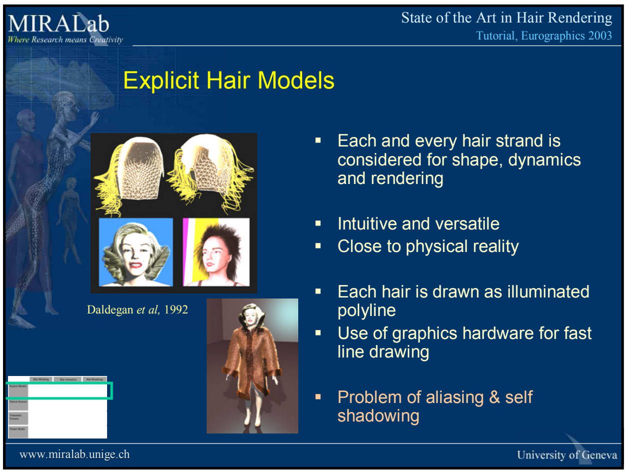
MIRALab
Where Research means Creativity

State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering and Explicit Models

	Hair Modeling	Hair Animation	Hair Rendering
Explicit Models	effective *- tedious to model* *- not suitable for knots and braids*	adequate *- expensive due to size* *- inappropriate for hair-hair interaction*	fast *- inadequate for self-shadowing*
Particle Systems	inappropriate	adhoc *- lacks physical basis* *- no hair-hair interaction*	effective *- lacks shadowing and self-shadowing*
Volumetric Textures	effective *- not suitable for long hair*	limited *- via Animated Shape Perturbation*	effective *- expensive*
Cluster Model	effective *- not suitable for simple smooth hair*	not done *- via Animated Shape Perturbation*	effective

www.miralab.unige.ch
University of Geneva



MIRALab
Where Research means Creativity

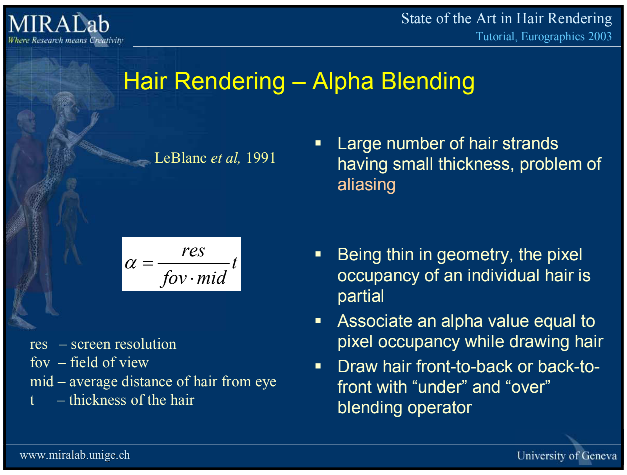
State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Explicit Hair Models

Daldegan *et al*, 1992

- Each and every hair strand is considered for shape, dynamics and rendering
- Intuitive and versatile
- Close to physical reality
- Each hair is drawn as illuminated polyline
- Use of graphics hardware for fast line drawing
- Problem of aliasing & self shadowing

www.miralab.unige.ch
University of Geneva



MIRALab
Where Research means Creativity

State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Alpha Blending

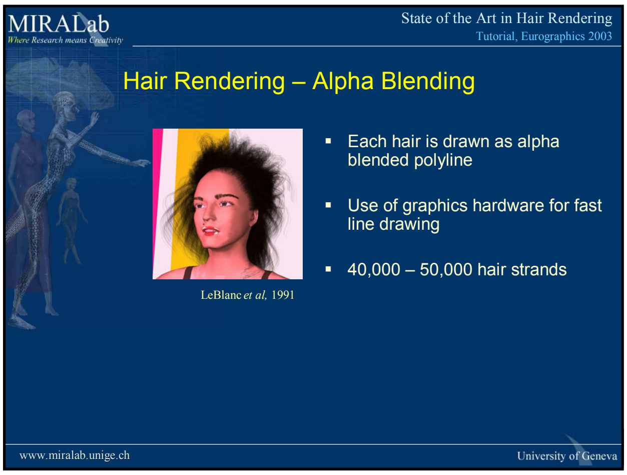
LeBlanc *et al*, 1991

$$\alpha = \frac{res}{fov \cdot mid} t$$

res – screen resolution
fov – field of view
mid – average distance of hair from eye
t – thickness of the hair

- Large number of hair strands having small thickness, problem of aliasing
- Being thin in geometry, the pixel occupancy of an individual hair is partial
- Associate an alpha value equal to pixel occupancy while drawing hair
- Draw hair front-to-back or back-to-front with “under” and “over” blending operator

www.miralab.unige.ch
University of Geneva



MIRALab
Where Research means Creativity

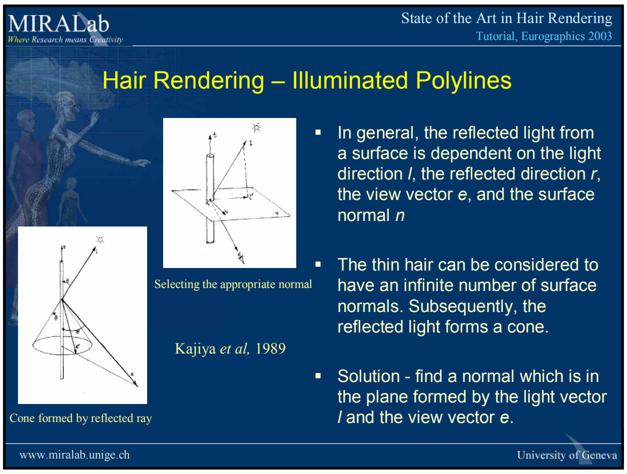
State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Alpha Blending

LeBlanc *et al*, 1991

- Each hair is drawn as alpha blended polyline
- Use of graphics hardware for fast line drawing
- 40,000 – 50,000 hair strands

www.miralab.unige.ch
University of Geneva



MIRALab
Where Research means Creativity

State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Illuminated Polylines

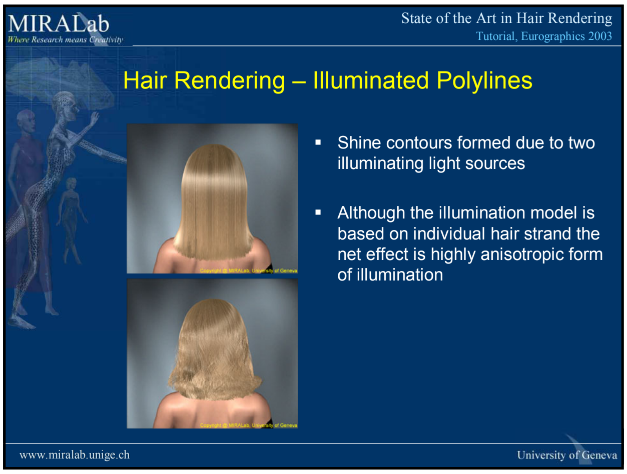
Selecting the appropriate normal

Cone formed by reflected ray

Kajiya *et al*, 1989

- In general, the reflected light from a surface is dependent on the light direction *l*, the reflected direction *r*, the view vector *e*, and the surface normal *n*
- The thin hair can be considered to have an infinite number of surface normals. Subsequently, the reflected light forms a cone.
- Solution - find a normal which is in the plane formed by the light vector *l* and the view vector *e*.

www.miralab.unige.ch
University of Geneva



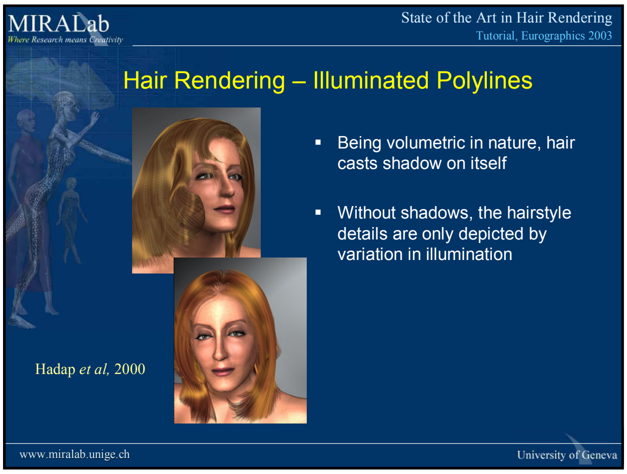
MIRALab
Where Research means Creativity

State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Illuminated Polylines

- Shine contours formed due to two illuminating light sources
- Although the illumination model is based on individual hair strand the net effect is highly anisotropic form of illumination

www.miralab.unige.ch
University of Geneva



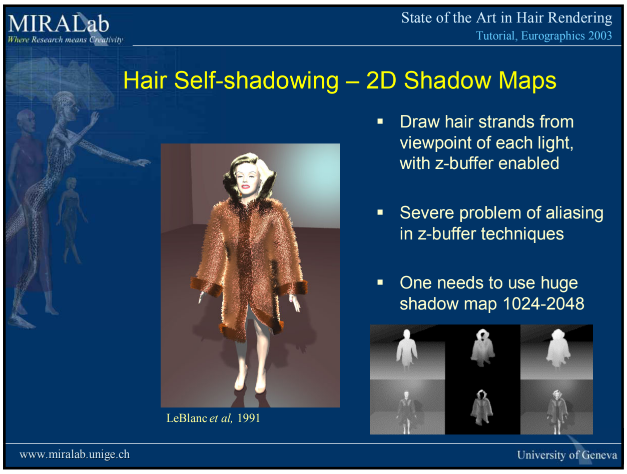
State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Illuminated Polylines

- Being volumetric in nature, hair casts shadow on itself
- Without shadows, the hairstyle details are only depicted by variation in illumination

Hadap *et al*, 2000

www.miralab.unige.ch University of Geneva



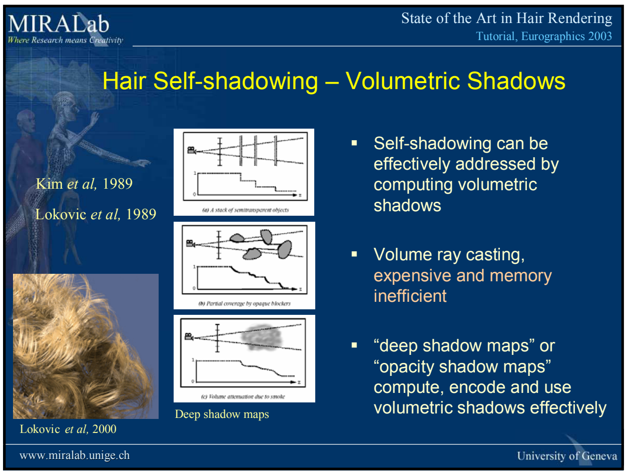
State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Self-shadowing – 2D Shadow Maps

- Draw hair strands from viewpoint of each light, with z-buffer enabled
- Severe problem of aliasing in z-buffer techniques
- One needs to use huge shadow map 1024-2048

LeBlanc *et al*, 1991

www.miralab.unige.ch University of Geneva



State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Self-shadowing – Volumetric Shadows

Kim *et al*, 1989

Lokovic *et al*, 1989

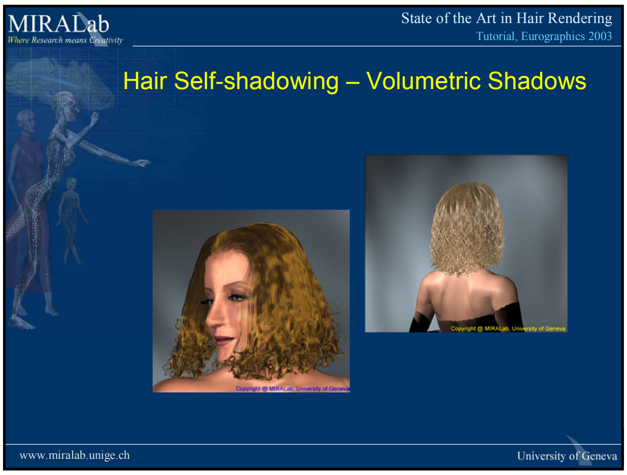
(a) A stack of semitransparent objects
(b) Partial coverage by opaque blockers
(c) Volume attenuation due to smoke

Deep shadow maps

Lokovic *et al*, 2000

- Self-shadowing can be effectively addressed by computing volumetric shadows
- Volume ray casting, expensive and memory inefficient
- "deep shadow maps" or "opacity shadow maps" compute, encode and use volumetric shadows effectively

www.miralab.unige.ch University of Geneva

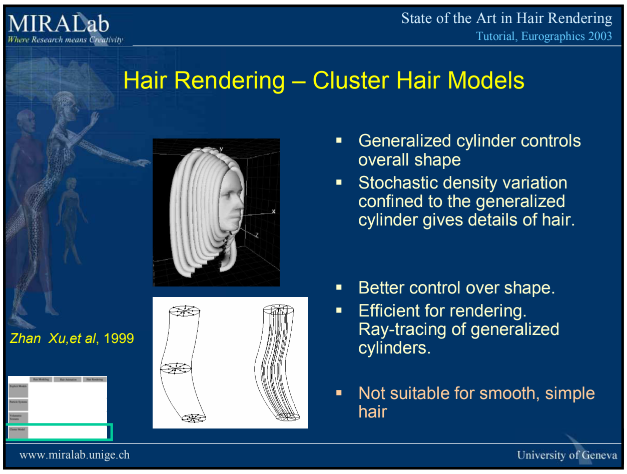


State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Self-shadowing – Volumetric Shadows

Copyright @ MIRALab, University of Geneva
Copyright @ MIRALab, University of Geneva

www.miralab.unige.ch University of Geneva



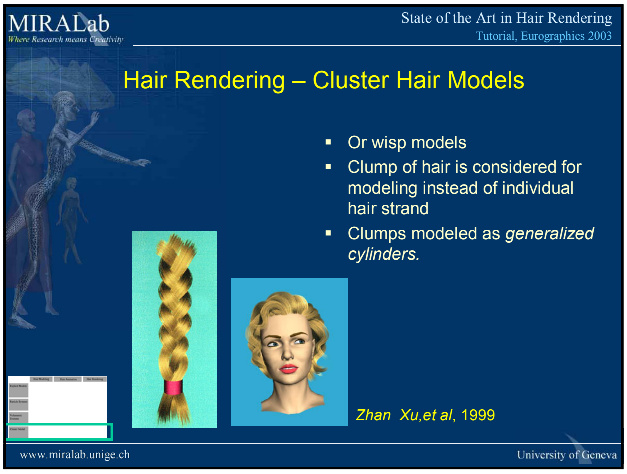
State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Cluster Hair Models

- Generalized cylinder controls overall shape
- Stochastic density variation confined to the generalized cylinder gives details of hair.
- Better control over shape.
- Efficient for rendering. Ray-tracing of generalized cylinders.
- Not suitable for smooth, simple hair

Zhan Xu,et al, 1999

www.miralab.unige.ch University of Geneva



State of the Art in Hair Rendering
Tutorial, Eurographics 2003

Hair Rendering – Cluster Hair Models

- Or wisp models
- Clump of hair is considered for modeling instead of individual hair strand
- Clumps modeled as *generalized cylinders.*

Zhan Xu,et al, 1999

www.miralab.unige.ch University of Geneva

State of the Art in Hair Simulation

Nadia Magnenat-Thalmann Sunil Hadap Prem Kalra *

MIRALab, CUI, University of Geneva
24, rue du General Dufour, CH-1211 Geneva, Switzerland
email: {thalmann,sunil}@cui.unige.ch

Abstract

In this paper we summarize the technological advances in hair simulation for computer graphics. There are mainly three tasks in hair simulation - Hair Shape Modeling, Hair Dynamics and Hair Rendering. Various models developed for these tasks, fall mainly in the categories of *particle systems*, *explicit hair models*, *cluster hair models* and models based on *volumetric textures*. We discuss advantages and disadvantages of each of these approaches. We also introduce a new hair shape modeling paradigm based on fluid flow. The proposed method provides a sound basis for modeling hair-body and hair-hair interaction.

Keywords: hair shape modeling, hair animation, hair rendering, hypertexture

1 Introduction

One of the many challenges in simulating believable virtual humans has been to produce realistic looking hair. The virtual humans, two decades ago, were given polygonal hair structure. Today, this is not acceptable. Realistic visual depiction of virtual humans has improved over the years. Attention has been given to all the details necessary for producing visually convincing virtual humans and many improvements have been done to this effect.

On a scalp, human hair are typically 100,000 to 150,000 in number. Geometrically they are long thin curved cylinders having varying thickness. The strands of hair can have any degree of waviness from straight to curly. The hair color can change from white to grey, red to brown, due to the pigmentation, and have shininess. Thus, difficulties of simulating hair stem from the huge number and geometric intricacies of individual hair, complex interaction of light and shadow among the hairs, the small scale of thickness of one hair compared to the rendered image and intriguing hair to hair interaction while in motion. One can conceive three main aspects in hair simulation - hair shape modeling, hair dynamics or animation, and hair rendering. Often these aspects are interconnected while processing hairs. Hair shape modeling deals with exact or fake creation of thousands of individual hair - their geometry, density, distribution, and orientation. Dynamics of hair addresses hair movement, their collision with other objects particularly relevant for long hair, and self-collision of hair. The rendering of hair involves dealing with hair color, shadow, specular highlights, varying degree of transparency and anti-aliasing. Each of the aspects is a topic of research.

Many research efforts have been done in hair simulation research, some dealing only with one of the aspects of simulation -shape modeling, dynamics or rendering. Several research efforts were inspired by the general problem of simulation of natural phenomena such as grass, and trees. These addressed a more limited problem of simulating of fur or short hair. We divide hair simulation models into four categories depending upon the underlying technique involved: *particle systems, volumetric textures, explicit hair models and cluster hair model*. We discuss models presented by researchers in each of these model categories and state their contribution to the three aspects of hair simulation, i.e. hair shape modeling, hair dynamics and hair rendering. We also introduce a new hair shape modeling paradigm based on fluid flow.

The paper is organized as follows. First we give the state of the art in hair shape modeling. The hair shape modeling research in each category of the simulation models is presented. Models for hair dynamics are briefly described in Section 3. Section 4 presents the problem of hair rendering and the various solutions proposed by different people. Finally, we summarize the effectiveness and limitations of models in the four categories related to each aspect of hair simulation in the form of a table. Some future avenues for research in hair simulation are also outlined.

2 Hair Shape Modeling

Intricate hairstyle is indeed a consequence of physical properties of an individual hair and complex hair-hair and hair-body interactions. As we will see in the next section, modeling complex hair dynamics, that too at interactive speeds, is currently impractical. For the reasons, it would be worthwhile to treat *hair shape modeling* as a separate problem and use some heuristic approach.

Early attempts of styling long hair were based on *explicit hair models*. In the explicit hair model, each hair strand is considered for the shape and the dynamics. Daldegan

[*] Visiting from Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India. pkalra@cse.iitd.ernet.in

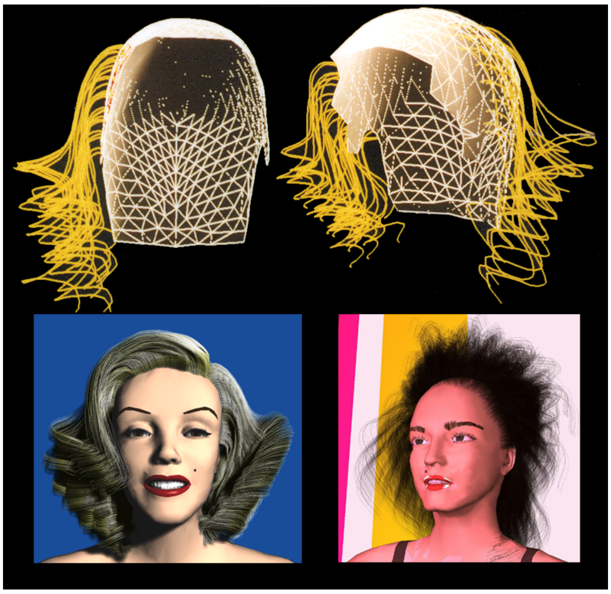


Figure 1: Hairstyling by defining a few curves in 3D

et al [5] proposed that the user could interactively define a few characteristic hair strands in 3D and then populate the hair style based on them. The user is provided with a flexible graphical user interface to sketch a curve in 3D around the scalp. A few parameters such as density, spread, jitter and orientation control the process that duplicates the characteristic hairs to form a hair style. Figure 1 illustrates the method of defining few characteristic curves and resulting hairstyles from the method. Similarly, even for the fur modeling, Daldegan *et al* [4], Gelder *et al* [8] and Bruderlin *et al* [1] took similar explicit hair modeling approach. Figure 12 illustrates a furry coat modeled by the explicit hair model.

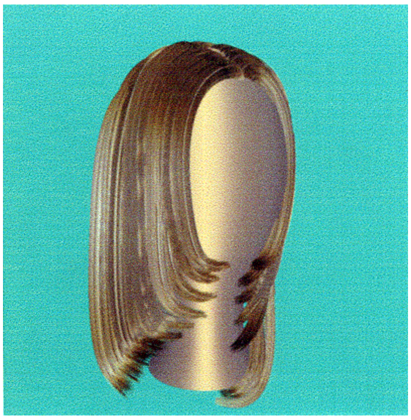


Figure 2: Cluster Hair Model, by Yan *et al*

The explicit hair models are very intuitive and close to reality. Unfortunately, they are tedious for hairstyling. Typically, it takes 5-10 hours to model a complex hair style, as in figure 1, using the method in [5]. They are also numerically expensive for hair dynamics. These difficulties are partially overcome by considering a bunch of hair instead of individual hair in the case of the wisp/cluster models. This assumption is quite valid as in reality. Due to effects of adhesive/cohesive forces, hairs tend to form clumps. Watanabe introduced the wisp modeled in [24, 25]. Yan *et al* [26] modeled the wisps as *generalized cylinders*, see figure 2. One of the contributions of the work was also in rendering of hair using the blend of ray-tracing generalised cylinders and the *volumetric textures*. The wisp model is also evident in [2]. Surprisingly, till now, the wisp models are only limited to static hair shape modeling and we feel that it offers an interesting research possibility of modeling hair dynamics, efficiently. It would be interesting to model, how hair leave one wisp and join the other under dynamics.

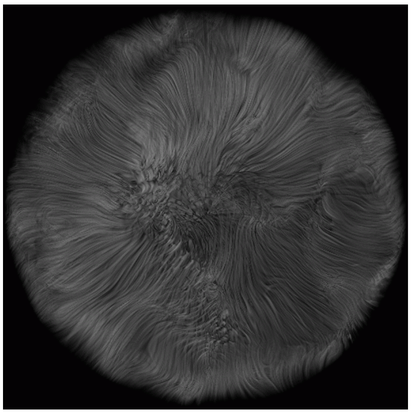


Figure 3: Fur as a Volumetric Texture, by Perlin *et al*

Nature exhibits some interesting fuzzy objects such as clouds, fire, eroded rocks and fur for which it is hard to have explicit geometric definition. Using the volumetric texture approach, fur can be modeled as a volumetric density function. Perlin *et al* [18] introduced *hypertextures*, which can model fur, see figure 3. Here, fur is modeled as intricate density variations in a 3D space, which gives an illusion of the fur like medium without defining geometry of each and every fiber. The model is essentially an extension to procedural solid texture synthesis evaluated through out the region, instead of only in the surface. They demonstrated that, combinations of simple analytical functions could define furry ball or furry donut. They further used 3D vector valued noise and turbulence to perturb the 3D texture space. This gave the natural looks to the otherwise even fur defined by the hypertexture. A good discussion on the procedural approach to modeling volumetric texture and fur in particular is in [7]. Hypertexture method by Perlin *et al* is only limited to geometries that can be analytically defined. Kajiya *et al* [12] extended this approach to have hypertextures tiled on to complex geometry. They demonstrated this by modeling a furry bear, see figure 10. They used a single solid texture tile namely texel and mapped it repeatedly on the bear's geometry. The texels automatically orient in the direction away from the surface and thus one has fuzzy volumetric density variation around the bear, which is the fur.

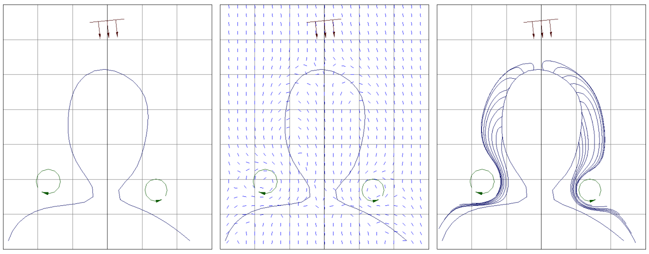


Figure 4: Hair as streamlines of a fluid flow

As evident from previous discussions, one of the strengths of the explicit hair models is their intuitiveness and ability to control the global shape of the hair. On the contrary, volumetric textures give a nice way of interpreting complexity in nature and they are rich in details. We notice that the fluid flow has both the characteristics, which we would like to exploit for hair shape modeling. We model hair shape as streamlines of a fluid flow. For complete details of the method, we refer to [10]. We choose the flow to be an ideal flow. User can setup few ideal flow elements around the body geometry to design a hairstyle, as shown in figure 2. The hair-body interaction is modeled using *source panel method* and hair-hair interaction is handled by the continuum property of fluid. Thus user can design complex hairstyles without worrying about hair-body and hair-hair interaction. Hairstyles in figure 5 and 6 are the examples of modeling hair as a fluid flow.

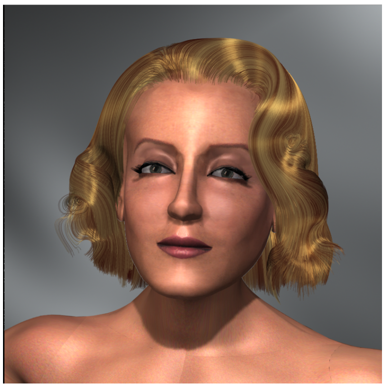


Figure 5: Hair as a fluid flow

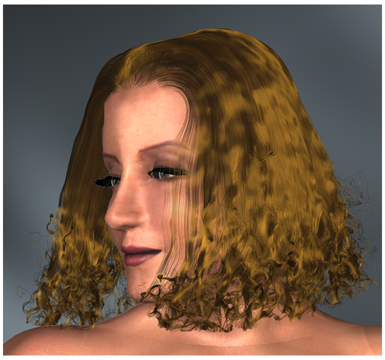


Figure 6: Adding overall volumetric perturbations to the fluid flow

3 Hair Dynamics

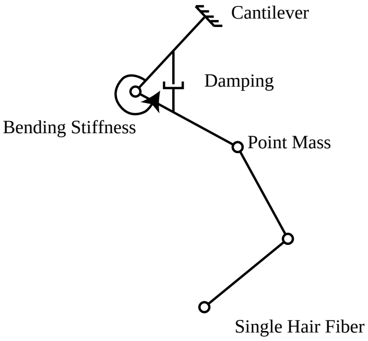


Figure 7: Simple mass-spring system for an individual hair dynamics

Anjyo *et al* [11], Rosenblum *et al* [22] and Kurihara *et al* [23] developed dynamic models that are issentialy based on individual hair. An individual hair is modeled as connected rigid segments having bending stiffness at each joint. Then the individual hair is solved for the movement due to the inertial forces and the collision with the body. Though the cantilever dynamics and collision avoidance with the body of each hair is within the scope of current computing power, modeling complex hair-to-hair interaction is still a challenge. Figure 8 illustrates the effectiveness of the dynamic model even though no hair-hair interaction is considered.

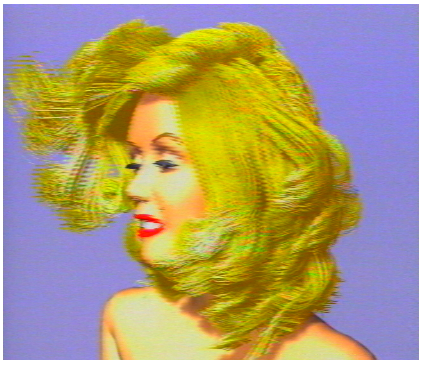


Figure 8: Hair animation using the explicit model, by Kurihara *et al*

In the case of fur, which is mostly modeled as volumetric texture, one cannot take the explicit model approach for the animation. In this case, a time varying volume density function can facilitate animation of fur. One can simulate effects of turbulent air on the fur using stochastic space perturbation such as turbulence, noise, Brownian motion etc. Apart from Lewis [15] and Perlin [17, 18], work by Dischler [6] gave a generalized method for these animated shape perturbations.

4 Hair Rendering

In the field of virtual humans, hair presents one of the most challenging rendering problems. The difficulties arise from various reasons: large number of hair, detailed geometry of individual hair and complex interaction of light and shadow among the hairs and their small thickness. The rendering of hair often suffers from the aliasing problem due to many individual hairs reflecting light and casting shadows on each other contribute to the shading of each pixel. Further, concerning display of hairs, we see not only individual hairs but also a continuous image consisting of regions of hair color, shadow, specular highlights, varying degree of transparency and haloing under backlight conditions. The image, in spite of the structural complexity, shows a definite pattern and texture in its aggregate form.

In the last decade, the hair-rendering problem has been addressed by a number of researchers, in some cases with considerable success. However, most cases work well in particular conditions and offer limited (or none) capabilities in terms of dynamics or animation of hair. Much of the work refers to a more limited problem of rendering fur, which also has a lot in common with rendering natural phenomena such as grass and trees. As follows we give the related work in hair rendering focusing their salient features and limitations.

Particle systems introduced by Reeves *et al* [19], primarily meant to model class of fuzzy objects such as fire. Despite particles small size -smaller than even a pixel- the particle manifests itself by the way it reflects light, casts shadows, and occludes objects. Thus, the subpixel structure of the particle needs to be represented only by a model that can represent these properties. A particle system is rendered by painting each particle in succession onto the frame buffer, computing its contribution to the pixel and compositing it to get the final color at the pixel. The technique has been successfully used for rendering these fuzzy objects and integrated in many commercial animation systems. Figure 9 is an example of how one can use connected particle systems for the modeling of hair. However, the technique has some limitations for shadowing and self-shadowing. Much of it is due to the inherent modeling using particle systems: simple stochastic models are not adequate to represent the type of order and orientation of hair. Also, it requires appropriate lighting model to capture and control the hair length and orientation. The specular highlights in particular owing to the geometry of the individual strands are highly anisotropic.

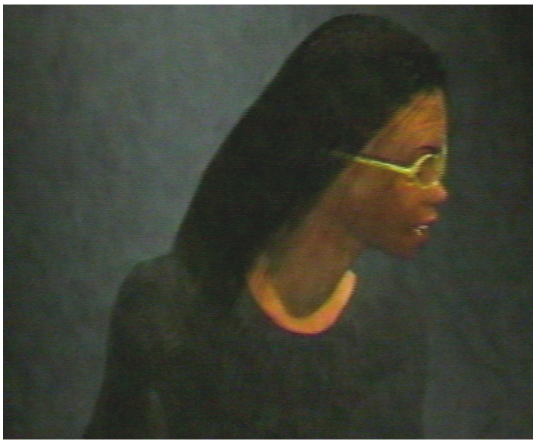


Figure 9: Hair as Connected Particle System, “The End” by Alias—Wavefront

Impressive results have been obtained for the more limited problem of rendering fur, which can be considered as very short hair. As we have already discussed in the case of hair shape modeling, Perlin *et al* [18] introduced hypertextures that can model fur like objects. Hypertexture approach remains limited to geometries that can be defined analytically. Kajiya and Kay extended this approach to use it on the complex geometries. They used a single solid texture tile namely texel. The idea of texels was inspired by the notion of volume density used in [18]. A texel is a 3D texture map where both the surface frame and lighting model parameters are embedded over a volume. Texels are a type of model intermediate between a texture and a geometry. A texel is however, not tied to the geometry of any particular surface and thus makes the rendering time independent of the geometric complexity of the surface that it extracts. The results are demonstrated by rendering a teddy bear (figure 10). Texels are rendered using ray casting, in a manner similar to that for volume densities using a suitable illumination model. Kajiya *et al* discusses more about the particular fur illumination model and a general rendering method for rendering volume densities. The rendering of volume densities are also covered in great detail in the book by Eber *et al* [7].

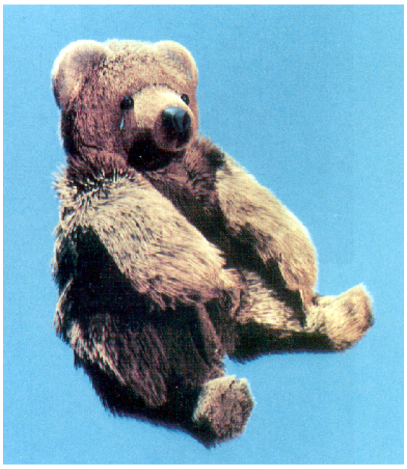


Figure 10: Volumetric Texture rendering by Kajiya *et al*

In another approach by Goldman [9], emphasis is given on rendering visual characteristics of fur in cases where the hair geometry is not visible at the final image resolution -object being far away from the camera. A probabilistic rendering algorithm, also referred to as fakefur algorithm is proposed. In this model, the reflected light from individual hairs and from the skin below is blended using the expectations of a ray striking a hair in that area as the opacity factor.

Though the volumetric textures are quite suitable for rendering furry objects or hair patches, rendering of long hair using this approach does not seem obvious.

A brute force method to render hair is to model each individual hair as curved cylinder and render each cylinder primitive. The shear number of primitives modeling hair poses serious problem to this approach. However, the explicit modeling of hair has been used for different reasons employing different types of primitives.

An early effort by Csuri *et al* [3] generated fur-like volumes using polygons. Each hair was modeled as a single triangle laid out on a surface and rendered using a Z-buffer algorithm for hidden surface removal. Miller [16] produced better results by modeling hair as pyramids consisting of triangles. Oversampling was employed for anti-aliasing. These techniques however, impose serious problems considering reasonable number and size of hairs.

In an another approach, a hardware Z-buffer renderer was used with Gouraud shading for rendering hair modeled as connected segments of triangular prisms on a full human head. However, the illumination model used was quite simplistic and no effort was done to deal with the problem of aliasing. LeBlanc *et al* [14] proposed an approach of rendering hair using pixel blending and shadow buffers. This technique has been one of the most effective and practical hair rendering approach. Though it could be applied for the variety of hairy and furry objects, one of the primary intention of the approach was to be able to render realistic different styles of human hairs. Hair rendering is done by mix of ray tracing and drawing polyline premitives, with added module for the shadow buffer [20]. The rendering pipeline has the following steps: first the shadow of the scene is calculated for each light source. Then, hair shadow buffer is computed for each light source for the given hair style model; this is done by drawing each hair segment into a Z-buffer and extracting the depth map. The depth maps for the shadow buffers for the scene and hair are composed giving a single composite shadow buffer for each light source. The scene image with its Z-buffer is generated using scene model and composite shadow buffers. The hair segments are then drawn as illuminated polylines [27] into the scene using Z-buffer of scene for determining the visibility and the composite shadow buffers for finding the shadows. Figure 11 shows the process and Figure 12 gives final rendered image of a hairstyle of a synthetic actor with a fur coat.

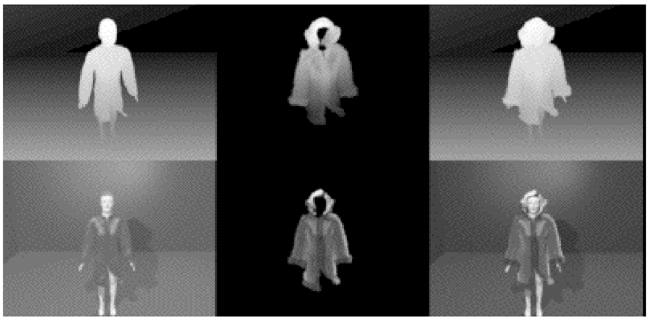


Figure 11: Rendering pipeline of the method-”Pixel Blending and Shadow Buffer”

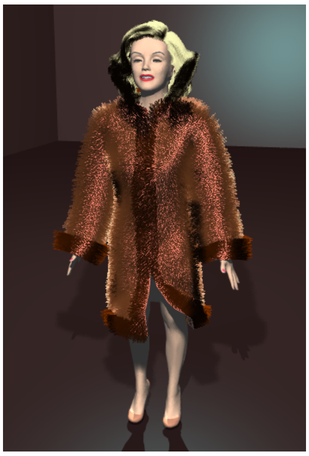


Figure 12: Fur using Explicit Hair Model

Special effects like rendering wet hair require change in the shading model. Bruderlin [1] presented some simple ways to account for the wetness of hair -changing the specularity. That is, hairs on the side of a clump facing the light are brighter than hairs on a clump away from the light.

Kong and Nakajima *et al* [13] presented an approach of using visible volume buffer to reduce the rendering time. The volume buffer is a 3D cubical space defined by the user depending upon the available memory and the resolution required. They consider hair model as combination coarse background hair and detailed surface hair determined by the distance from the viewpoint or the opacity value. The technique reduces considerably the rendering time, however, the quality of results is not so impressive.

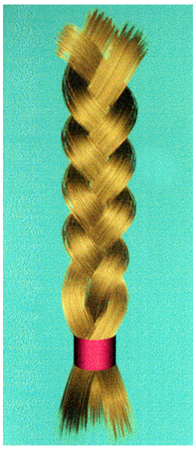


Figure 13: Braid rendered using generalized cylinders and volumetric texture, by Yan *et al*

Yan *et al* [26] combine volumetric texture inside the explicit geometry of hair cluster defined as a generalized cylinder. Ray tracing is employed to get the boundaries of the generalized cylinder and then the standard volume rendering is applied along the ray to capture the characteristics of the density function defined. This may be considered as a hybrid approach for hair rendering.

5 Conclusion

	Hair Modeling	Hair Animation	Hair Rendering
Explicit Models	effective - *tedious to model* - *not suitable for knots and braids*	adequate - *expensive due to size* - *inappropriate for hair-hair interaction*	fast - *inadequate for self-shadowing*
Particle Systems	inappropriate	adhoc - *lacks physical basis* - *no hair-hair interaction*	effective - *lacks shadowing and self-shadowing*
Volumetric Textures	effective - *not suitable for long hair*	limited - *via Animated Shape Perturbation*	effective - *expensive*
Cluster Model	effective - *not suitable for simple smooth hair*	not done - *via Animated Shape Perturbation*	effective
Hair as a Fluid	effective - *not suitable for knots and braids*	not done	not done

Figure 14: Comparison of the various hair models

In this paper we present the state of the art in hair simulation, one of the most challenging problem of virtual humans. We consider three aspects in hair simulation: hair shape modeling, hair rendering and hair dynamics. Different approaches have been proposed in the literature dealing with one or more aspects of hair simulation. We divide them into four categories based on the underlying technique: particle systems, volumetric textures, explicit hair models and cluster hair model. Some of these techniques are appropriate and effective only for one of the aspects in hair simulation. In figure 14, we summarize their role with their effectiveness and limitations for each aspect of the hair simulation. Notice that we have introduced a new hair modeling paradigm - “Hair as a Fluid”. We believe, this approach has good potential in terms of hair shape modeling and hair dynamics, as the methodology gives a basis for modeling complex hair-hair interactions.

No, doubt research in hair simulation despite the inherent difficulty of its size has been encouraging and shown remarkable improvements over the years. People in general are not ready to accept a bald digital actor or an animal without fur. Such realism to computer graphics characters is also becoming more widely available to the animators. Many of the commercial software provide suitable solutions and plug ins for creating hairy and furry characters. An article by Robertson [21] gives an overview of various techniques available for animators.

However, the quest of realism increases after noticing what one can already achieve. This asks to continue our research for better solutions. Hair dynamics for instance remains an area, where existing computing resources impose constraints. It is still very far to imagine real time hair blowing with full rendering and collisions. Hair dressing and styling also require flexible and convenient modeling paradigms. Fast and effective rendering methods for all hair styles -short or long, in all conditions -dry or wet, modeling all the optical properties of hair are still to be explored. So there is still long way to go.

6 Acknowledgements

This work is supported by the Swiss National Research Foundation (FNRS).

References

[1] Bruderlin, A. A method to generate wet and broken-up animal fur. *Pacific Graphics ’99* (October 1999). Held in Seoul, Korea.

[2] Chen, L.-H., Saeyor, S., Dohi, H., and Ishizuka, M. A system of 3d hair style synthesis based on the wisp model. *The Visual Computer 15*, 4 (1999), 159–170. Springer-Verlag, ISSN 0178-2789.

[3] CSURI, C., HAKATHORN, R., AND PARENT, R. Towards an interactive high visual complexity animation system. In *Computer Graphics* (1979).

[4] DALDEGAN, A., AND MAGNENAT-THALMANN, N. Creating virtual fur and hair styles for synthetic actors. In *Communicating with Virtual Worlds* (1993), N. Magnenat-Thalmann and D. Thalmann, Eds., Springer-Verlag.

[5] DALDEGAN, A., THALMANN, N. M., KURIHARA, T., AND THALMANN, D. An integrated system for modeling, animating and rendering hair. *Computer Graphics Forum (Eurographics '93) 12*, 3 (1993), 211–221. Held in Oxford, UK.

[6] DISCHLER, J.-M. A general model of animated shape perturbation. *Graphics Interface '99* (June 1999), 140–147. ISBN 1-55860-632-7.

[7] EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K., AND WORLEY, S. *Texturing and Modeling*. Academic Press, 1998.

[8] GELDER, A. V., AND WILHELMS, J. An interactive fur modeling technique. *Graphics Interface '97* (May 1997), 181–188. ISBN 0-9695338-6-1 ISSN 0713-5424.

[9] GOLDMAN, D. B. Fake fur rendering. *Proceedings of SIGGRAPH 97* (August 1997), 127–134. ISBN 0-89791-896-7. Held in Los Angeles, California.

[10] HADAP, S., AND MAGNENAT-THALMANN, N. Interactive hair styler based on fluid flow. In *Proceedings of Eurographics Workshop on Computer Animation and Simulation '2000* (2000). to appear.

[11] ICHI ANJYO, K., USAMI, Y., AND KURIHARA, T. A simple method for extracting the natural beauty of hair. *Computer Graphics (Proceedings of SIGGRAPH 92) 26*, 2 (July 1992), 111–120. ISBN 0-201-51585-7. Held in Chicago, Illinois.

[12] KAJIYA, J. T., AND KAY, T. L. Rendering fur with three dimensional textures. *Computer Graphics (Proceedings of SIGGRAPH 89) 23*, 3 (July 1989), 271–280. Held in Boston, Massachusetts.

[13] KONG, W., AND NAKAJIMA, M. Visible volume buffer for efficient hair expression and shadow generation. *Computer Animation '99, IEEE Computer Society* (May 1999). IEEE Press, Held in Geneva, Switzerland.

[14] LEBLANC, A., TURNER, R., AND THALMANN, D. Rendering hair using pixel blending and shadow buffer. *Journal of Visualization and Computer Animation 2* (1991), 92–97. John Wiley.

[15] LEWIS, J.-P. Algorithms for solid noise synthesis. *Computer Graphics (Proceedings of SIGGRAPH 89) 23*, 3 (July 1989), 263–270. Held in Boston, Massachusetts.

[16] MILLER, G. S. P. From wire-frames to furry animals. *Graphics Interface '88* (June 1988), 138–145.

[17] PERLIN, K. An image synthesizer. *Computer Graphics (Proceedings of SIGGRAPH 85) 19*, 3 (July 1985), 287–296. Held in San Francisco, California.

[18] PERLIN, K., AND HOFFERT, E. M. Hypertexture. *Computer Graphics (Proceedings of SIGGRAPH 89) 23*, 3 (July 1989), 253–262. Held in Boston, Massachusetts.

[19] REEVES, W. T. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics 2*, 2 (April 1983), 91–108. Held in USA.

[20] REEVES, W. T., SALESIN, D. H., AND COOK, R. L. Rendering antialiased shadows with depth maps. *Computer Graphics (Proceedings of SIGGRAPH 87) 21*, 4 (July 1987), 283–291. Held in Anaheim, California.

[21] ROBERTSON, B. Hair-raising effects. *Computer Graphics World, Magazine* (October 1995).

[22] ROSENBLUM, R., CARLSON, W., AND TRIPP, E. Simulating the structure and dynamics of human hair: Modeling, rendering and animation. *Journal of Visualzation and Computer Animation 2* (June 1991), 141–148. John Wiley.

[23] TSUNEYA KURIHARA, KEN-ICHI ANJYO, D. T. *Models and Techniques in Computer Animation*. Springer-Verlag, ch. Hair Animation with Collision Detection.

[24] WATANABE, Y., AND SUENAGA, Y. Drawing human hair using wisp model. In *Proceedings of Computer Graphics International'89* (1989), Springer Verlag, pp. 691–700.

[25] WATANABE, Y., AND SUENAGA, Y. A trigonal prism-based method for hair image generation. *IEEE Computer Graphics & Applications 12*, 1 (January 1992), 47–53.

[26] YAN, X. D., XU, Z., YANG, J., AND WANG, T. The cluster hair model. *Journal of Graphics Models and Image Processing* (1999). Academic Press.

[27] ZCKLER, M., STALLING, D., AND HEGE, H.-C. Interactive visualization of 3d-vector fields using illuminated streamlines. *IEEE Visualization '96* (October 1996), 107–114. ISBN 0-89791-864-9.

EUROGRAPHICS 2003

Morphing and Caricatures

Volker Blanz

3D Scan Data

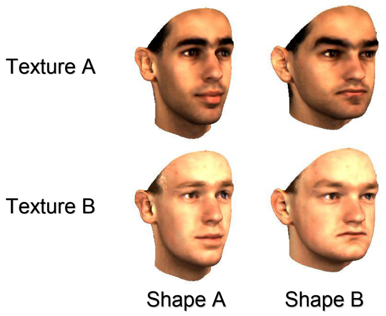
Face vectors derived from 3D scans

Manipulate faces in Face Space

- Separate Choice of Shape and Texture
- Morphing
- Caricaturing
- Learn attributes such as gender, weight and manipulate these

Exchange Shape and Texture

Shape and texture are correlated, but not strictly coupled.
Which faces are real scans?

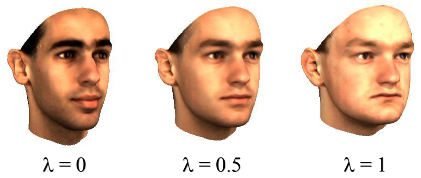


Morph between two faces

Faces $\mathbf{x}_1$, $\mathbf{x}_2$: $\mathbf{x}$ = shape, texture, or both.

$$\mathbf{x}(\lambda) = (1-\lambda)\mathbf{x}_1 + \lambda\mathbf{x}_2$$

Increase λ from 0 to 1



Caricatures

Exaggerate facial characteristics

Morph towards average:
features become less distinctive

Morph away from average:
features become more distinctive
= caricature

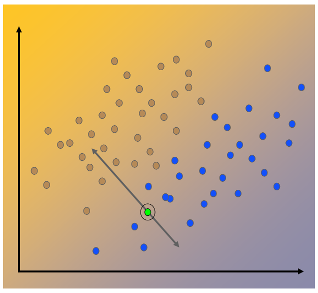
Gender, Body Weight, ...

Strategy:
Learn from examples.

Unlike expressions, need to investigate different individuals.

Separation of gender attributes from identity.

Labeled Examples



Linear Regression

Examples $\mathbf{x}_i$ with labels $y_i \in \{-1,1\}$ or $y_i \in [-1,1]$.

Find a mapping

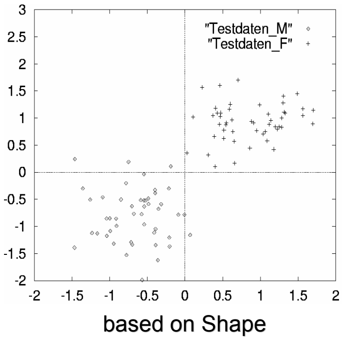
$$f : \Re^3 \to \Re, f(\mathbf{x}_i) = y_i$$

$$f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle \; + c$$

Gender Classification

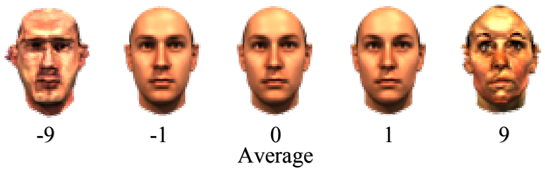
Classification of novel faces

based on Texture



Shift along ∇f

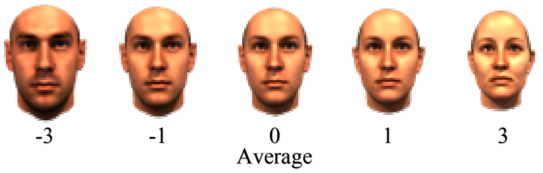
∇f : Steepest slope in L_2-Norm



Unsatisfactory! – Use statistics.

Shift along ∇f

∇f : Steepest slope in Mahalanobis distance (PCA)

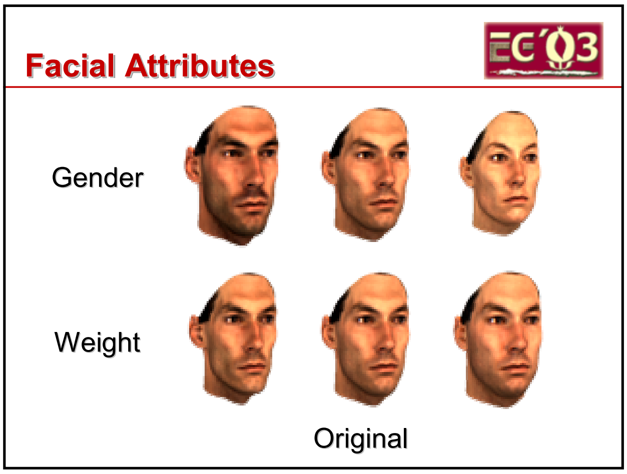


$\longrightarrow \quad \nabla f \; = \; \frac{1}{m}\sum_{i=1}^{m} y_i \mathbf{x}_i$ Weighted sum of examples.

Manipulation of Faces

Goal:

Manipulate gender, weight, …
But
Retain characteristics of the individual
such as mouth shape

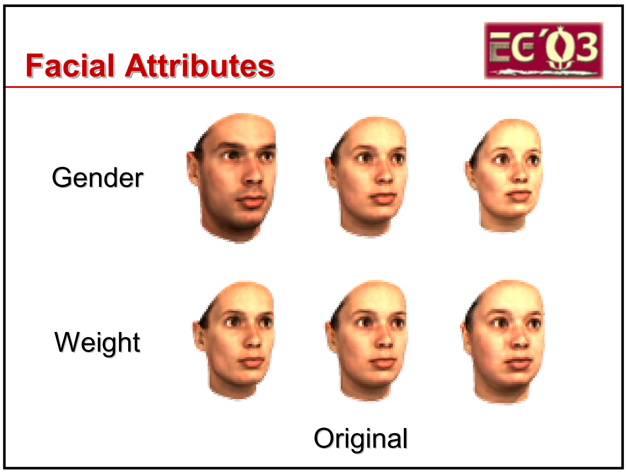


Facial Attributes

Gender

Weight

Original

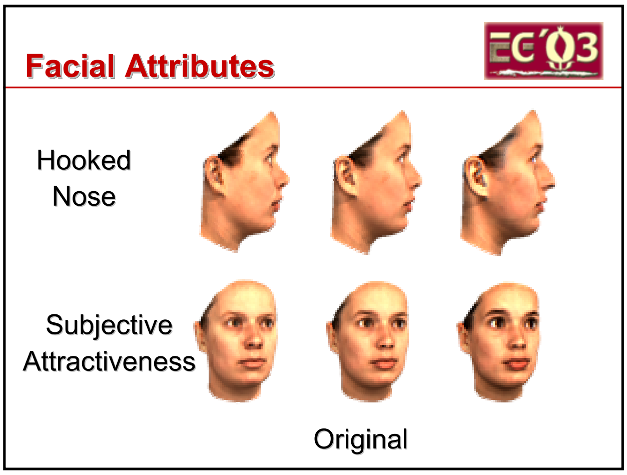


Facial Attributes

Gender

Weight

Original

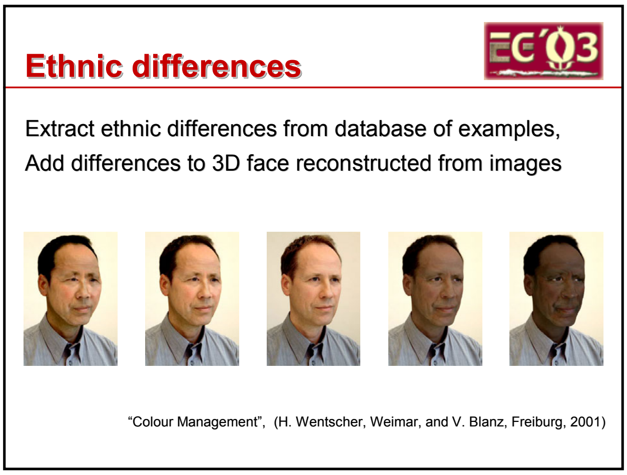


Facial Attributes

Hooked Nose

Subjective Attractiveness

Original



Ethnic differences

Extract ethnic differences from database of examples,
Add differences to 3D face reconstructed from images

“Colour Management”, (H. Wentscher, Weimar, and V. Blanz, Freiburg, 2001)

3. References

This section contains a list of publications, which are useful to learn more about facial modeling and animation. The selection of these publications represents the authors' point of view and might not be complete.

References

1. Irene Albrecht, Jörg Haber, Kolja Kähler, Marc Schröder, and Hans-Peter Seidel. *"May I talk to you? :-)"* — Facial Animation from Text. In Sabine Coquillart, Heung-Yeung Shum, and Shi-Min Hu, editors, *Proceedings of the Tenth Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2002)*, pages 77–86. IEEE Computer Society, October 2002.

2. Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. Automatic Generation of Non-Verbal Facial Expressions from Speech. In *Proceedings of Computer Graphics International 2002 (CGI 2002)*, pages 283–293, July 2002.

3. Ken-ichi Anjyo, Yoshiaki Usami, and Tsuneya Kurihara. A Simple Method for Extracting the Natural Beauty of Hair. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Conference Proceedings)*, volume 26, pages 111–120. ACM SIGGRAPH, July 1992.

4. Volker Blanz, Curzio Basso, Thomas Vetter, and Tomaso Poggio. Reanimating Faces in Images and Video. In P Brunet and D Fellner, editors, *Proceedings of EUROGRAPHICS 2003*, volume 22, Granada, Spain, 2003. Blackwell.

5. Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3D Faces. In Alyn Rockwood, editor, *Computer Graphics (SIGGRAPH '99 Conference Proceedings)*, pages 187–194. ACM SIGGRAPH, August 1999.

6. Christoph Bregler, Michele Covell, and Malcolm Slaney. Video Rewrite: Driving Visual Speech with Audio. In Turner Whitted, editor, *Computer Graphics (SIGGRAPH '97 Conference Proceedings)*, pages 353–360. ACM SIGGRAPH, August 1997.

7. Justine Cassell, Catherine Pelachaud, Norman Badler, Mark Steedman, Brett Achorn, Tripp Bechet, Brett Douville, Scott Prevost, and Matthew Stone. Animated Conversation: Rule–Based Generation of Facial Expression Gesture and Spoken Intonation for Multiple Converstaional Agents. In Andrew Glassner, editor, *Computer Graphics (SIGGRAPH '94 Conference Proceedings)*, pages 413–420. ACM SIGGRAPH, July 1994.

8. Lieu-Hen Chen, Santi Saeyor, Hiroshi Dohi, and Mitsuru Ishizuka. A System of 3D Hair Style Synthesis based on the Wisp Model. *The Visual Computer*, 15(4):159–170, 1999.

9. Michael M. Cohen and Dominic W. Massaro. Modeling Coarticulation in Synthetic Visual Speech. In Nadia M. Magnenat-Thalmann and Daniel Thalmann, editors, *Models and Techniques in Computer Animation*, pages 139–156. Springer–Verlag, 1993.

10. Agnes Daldegan, Nadia M. Magnenat-Thalmann, Tsuneya Kurihara, and Daniel Thalmann. An Integrated System for Modeling, Animating and Rendering Hair. In Roger J. Hubbold and Robert Juan, editors, *Computer Graphics Forum (Proceedings of Eurographics '93)*, volume 12, pages 211–221, September 1993.

11. Douglas DeCarlo, Dimitris Metaxas, and Matthew Stone. An Anthropometric Face Model using Variational Techniques. In Michael F. Cohen, editor, *Computer Graphics (SIGGRAPH '98 Conference Proceedings)*, pages 67–74. ACM SIGGRAPH, July 1998.

12. Tony Ezzat, Gadi Geiger, and Tomaso Poggio. Trainable Videorealistic Speech Animation. In John F. Hughes, editor, *ACM Transactions on Graphics (SIGGRAPH 2002 Conference Proceedings)*, pages 388–398. ACM SIGGRAPH, July 2002.

13. Taro Goto, Marc Escher, Christian Zanardi, and Nadia Magnenat-Thalmann. MPEG-4 based Animation with Face Feature Tracking. In Nadia Magnenat-Thalmann and Daniel Thalmann, editors, *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '99*, pages 89–98, 1999.

14. Hans Peter Graf, Eric Cosatto, and Tony Ezzat. Face Analysis for the Synthesis of Photo-Realistic Talking Heads. In *Proceedings 4th International Conference on Automatic Face and Gesture Recognition*, pages 189–194, 2000.

15. Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Frédéric Pighin. Making Faces. In Michael F. Cohen, editor, *Computer Graphics (SIGGRAPH '98 Conference Proceedings)*, pages 55–66. ACM SIGGRAPH, July 1998.

16. Sunil Hadap and Nadia Magnenat-Thalmann. Modeling Dynamic Hair as a Continuum. In Alan Chalmers and Theresa-Marie Rhyne, editors, *Computer Graphics Forum (Proceedings of Eurographics 2001)*, volume 20, pages C329–C338, September 2001.

17. Antonio Haro, Brian Guenter, and Irfan Essa. Real-time, Photo-realistic, Physically Based Rendering of Fine Scale Human Skin Structure. In Steven J. Gortler and Karol Myszkowski, editors, *Rendering Techniques 2001 (Proceedings 12th Eurographics Workshop on Rendering)*, pages 53–62, 2001.

18. Horace H. S. Ip and C. S. Chan. Script-Based Facial Gesture and Speech Animation Using a NURBS Based Face Model. *Computers & Graphics*, 20(6):881–891, November 1996.

19. Tomomi Ishii, Takami Yasuda, Shigeki Yokoi, and Jun-ichiro Toriwaki. A Generation Model for Human Skin Texture. In *Proceedings of Computer Graphics International '93*, pages 139–150. Springer–Verlag, 1993.

20. ISO/IEC. Overview of the MPEG-4 Standard. http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm, July 2000.

21. Won-Ki Jeong, Kolja Kähler, Jörg Haber, and Hans-Peter Seidel. Automatic Generation of Subdivision Surface Head Models from Point Cloud Data. In *Proceedings of Graphics Interface 2002*, pages 181–188. Canadian Human-Computer Communications Society, May 2002.

22. Kolja Kähler, Jörg Haber, and Hans-Peter Seidel. Geometry-based Muscle Modeling for Facial Animation. In *Proceedings of Graphics Interface 2001*, pages 37–46. Canadian Human-Computer Communications Society, June 2001.

23. Kolja Kähler, Jörg Haber, and Hans-Peter Seidel. Reanimating the Dead: Reconstruction of Expressive Faces from Skull Data. *ACM Transactions on Graphics (SIGGRAPH 2003 Conference Proceedings)*, 22(3), July 2003.

24. Kolja Kähler, Jörg Haber, Hitoshi Yamauchi, and Hans-Peter Seidel. Head shop: Generating animated head models with anatomical structure. In *Proceedings ACM SIGGRAPH Symposium on Computer Animation (SCA '02)*, pages 55–64, July 2002.

25. G. A. Kalberer, P. Mueller, and L. Van Gool. Biological motion of speech. In *Biologically Motivated Computer Vision*, volume 2525 of *Lecture Notes in Computer Science*, pages 199–206. Springer, 2002.

26. Prem Kalra, Angelo Mangili, Nadia Magnenat-Thalmann, and Daniel Thalmann. SMILE: A Multilayered Facial Animation System. In *Proceedings IFIP WG 5.10, Tokyo, Japan*, pages 189–198, 1991.

27. Won-Sok Lee and Nadia Magnenat-Thalmann. Fast Head Modeling for Animation. *Image and Vision Computing*, 18(4):355–364, March 2000.

28. Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Constructing Physics-based Facial Models of Individuals. In *Proceedings of Graphics Interface '93*, pages 1–8. Canadian Human-Computer Communications Society, May 1993.

29. Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic Modeling for Facial Animations. In Robert Cook, editor, *Computer Graphics (SIGGRAPH '95 Conference Proceedings)*, pages 55–62. ACM SIGGRAPH, August 1995.

30. John P. Lewis and Frederic I. Parke. Automated Lip-Synch and Speech Synthesis for Character Animation. In John M. Carroll and Peter P. Tanner, editors, *Proceedings of Human Factors in Computing Systems and Graphics Interface '87*, pages 143–147, April 1987.

31. Stephen R. Marschner, Brian Guenter, and Sashi Raghupathy. Modeling and Rendering for Realistic Facial Animation. In Bernard Péroche and Holly Rushmeier, editors, *Rendering Techniques 2000 (Proceedings 11th Eurographics Workshop on Rendering)*, pages 231–242, 2000.

32. Jun-yong Noh and Ulrich Neumann. A Survey of Facial Modeling and Animation Techniques. USC Technical Report 99-705, University of Southern California, Los Angeles, CA, 1999.

33. Jun-yong Noh and Ulrich Neumann. Expression Cloning. In Eugene Fiume, editor, *Computer Graphics (SIGGRAPH '01 Conference Proceedings)*, pages 277–288. ACM SIGGRAPH, August 2001.

34. Frederic I. Parke. Parameterized Models for Facial Animation. *IEEE Computer Graphics and Applications*, 2(9):61–68, November 1982.

35. Frederic I. Parke and Keith Waters, editors. *Computer Facial Animation*. A K Peters, Wellesley, MA, 1996.

36. Catherine Pelachaud, Norman Badler, and Mark Steedman. Generating Facial Expressions for Speech. *Cognitive Science*, 20(1):1–46, 1996.

37. Frédéric Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing Realistic Facial Expressions from Photographs. In Michael F. Cohen, editor, *Computer Graphics (SIGGRAPH '98 Conference Proceedings)*, pages 75–84. ACM SIGGRAPH, July 1998.

38. Stephen M. Platt and Norman I. Badler. Animating Facial Expressions. In *Computer Graphics (SIGGRAPH '81 Conference Proceedings)*, volume 15, pages 245–252. ACM SIGGRAPH, August 1981.

39. Marco Tarini, Hitoshi Yamauchi, Jörg Haber, and Hans-Peter Seidel. Texturing Faces. In *Proceedings of Graphics Interface 2002*, pages 89–98. Canadian Human-Computer Communications Society, May 2002.

40. Demetri Terzopoulos and Keith Waters. Physically-based Facial Modelling, Analysis, and Animation. *Journal of Visualization and Computer Animation*, 1(2):73–80, December 1990.

41. Allen Van Gelder. Approximate Simulation of Elastic Membranes by Triangulated Spring Meshes. *Journal of Graphics Tools*, 3(2):21–41, 1998.

42. Thomas Vetter and Volker Blanz. Estimating Coloured 3D Face Models from Single Images: An Example Based Approach. *Lecture Notes in Computer Science*, 1407:499–513, 1998.

43. Keith Waters. A Muscle Model for Animating Three-Dimensional Facial Expression. In *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, volume 21, pages 17–24. ACM SIGGRAPH, July 1987.

44. Keith Waters and Joe Frisbie. A Coordinated Muscle Model for Speech Animation. In *Proceedings of Graphics Interface '95*, pages 163–170. Canadian Human-Computer Communications Society, May 1995.

45. Keith Waters and Demetri Terzopoulos. Modeling and Animating Faces Using Scanned Data. *Journal of Visualization and Computer Animation*, 2(4):123–128, October–December 1991.

46. Lance Williams. Performance-Driven Facial Animation. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Conference Proceedings)*, volume 24, pages 235–242. ACM SIGGRAPH, August 1990.

47. Yin Wu, Prem Kalra, Laurent Moccozet, and Nadia Magnenat-Thalmann. Simulating Wrinkles and Skin Aging. *The Visual Computer*, 15(4):183–198, 1999.

48. Yin Wu, Nadia Magnenat-Thalmann, and Daniel Thalmann. A Plastic-Visco-Elastic Model for Wrinkles in Facial Animation and Skin Aging. In J. N. Chen, editor, *Proceedings of the Second Pacific Conference on Computer Graphics and Applications (Pacific Graphics '94)*, pages 201–214, August 1994.

4. Organizers and presenters

The tutorial is jointly organized and presented by:

Jörg Haber is a senior researcher at the Max-Planck-Institute for Computer Sciences in Saarbrücken, Germany. He received his Master's (1994) and PhD (1999) degrees in Mathematics from the Technische Universität München, Germany. During the last seven years he did research in various fields of computer graphics and image processing, including global illumination and real-time rendering techniques, scattered data approximation, and lossy image compression. For the last three years, his major research interests concentrate on modeling, animation, and rendering of human faces. He received the Heinz-Billing-Award of the Max-Planck-Society and the SaarLB Science Award for the design and implementation of a facial modeling and animation system.

Demetri Terzopoulos holds the Lucy and Henry Moses Professorship in the Sciences at New York University and is a professor of computer science and mathematics at NYU's Courant Institute. He is currently on leave from the University of Toronto where he is Professor of Computer Science and Professor of Electrical and Computer Engineering. He graduated from McGill University and received the PhD in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology (MIT). His published work includes more than 200 technical articles and several volumes, primarily in computer graphics and vision, and also in computer-aided design, medical imaging, artificial intelligence, and artificial life. He has given hundreds of invited talks around the world on these topics, including numerous keynote and plenary addresses. Terzopoulos is a Fellow of the IEEE. His many awards include computer graphics honors from the International Digital Media Foundation, Ars Electronica, and NICOGRAPH. The latter cited his research on human facial animation, which stems from his pioneering work on physics-based human facial modeling and deformable models.

Nadia Magnenat-Thalmann has pioneered research into virtual humans over the last 20 years, participating in and demonstrating some of the most spectacular state-of-the-art developments in the field, and is responsible for the rigorous and intensive academic research programs that made them possible. She studied at the University of Geneva and obtained several bachelor degrees including Psychology, Biology, and Computer Science. She received a MSc in Biochemistry in l972 and a PhD in Quantum Physics in 1977. From 1977 to 1989, she was a Professor at the University of Montreal in Canada. In l989, she founded MIRALab, an interdisciplinary creative research laboratory at the University of Geneva. She has been working since 15 years on simulating facial expressions, and is responsible for the first MPEG-4 software that has been given to the MPEG-4 committee. Actually, she is participating to several EU projects on the simulation of facial animation (SONG and INTERFACE), hair (MESH) and clothes (E-TAILOR).

Volker Blanz studied Physics at University of Tübingen, Germany, and University of Sussex, Brighton, UK. At Max-Planck Institute for biological Cybernetics, Tübingen, he wrote both his Diploma thesis (1995) on image-based object recognition, and his PhD thesis (2000) on automated reconstruction of 3D face models from images. In 1995, he worked on Multi-class Support Vector Machines at AT&T Bell Labs. In recent years, he has also worked for the Center for Biological and Computational Learning at MIT several times. From 2000 to 2003, he has been a research assistant at University of Freiburg, and since 2003, he is a researcher at Max-Planck Institute for Computer Sciences, Saarbrücken. His research interests are 3D face models, animation, face recognition, and learning theory.