ARTIC: Augmented Reality Tangible Interface by Color Evaluation

José Miguel Salles Dias
Miguel.Dias@iscte.pt

Nádia Jamal
Nadia.Jamal@iscte.pt

Pedro Silva
Pedro.Silva@iscte.pt

Rafael Bastos
Rafael.Bastos@iscte.pt

ADETTI/ISCTE, Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática, Edifício ISCTE, 1600-082 Lisboa, Portugal, www.adetti.iscte.pt

Abstract

We present ARTIC, a novel tangible interface that offers 3D user interaction which can be explored in the context of Augmented and Mixed Reality. Our system consists in tracking a portable artifact based on color evaluation by computer vision, and use it as a 3D input device. Using vision-based techniques, we are able to detect and track the motion of a physical object through color analysis and segmentation, and then perform virtual camera calibration. Using ARTIC, the user can easily manipulate the tangible artifact in order to interact, in the general sense, with 3D virtual objects. The system allows for 6DOF, providing a natural and simple experience of interactivity to the user, which is extremely important in a Tangible Augmented Reality approach.

Keywords

Augmented Reality (AR), Mixed Reality (MR), Tangible Interface, 6DOF, Color Segmentation, Camera Calibration, Motion Prediction

1. INTRODUCTION

The development of Augmented Reality in the last decade has promoted new forms of Human-Computer Interaction (HCI), namely Tangible Augmented Interfaces [Kato2001]. These interfaces explore new ways of interaction between the physical world and the virtual world, establishing a bridge between them.

By using physical objects as interfaces, it is possible to acquire a new level of freedom when interacting with digital information, allowing interaction to become easier. Having this in mind we have developed ARTIC, a tangible interface that is characterized mainly as being natural and non-intrusive and that offers 6 Degrees Of Freedom (6DOF) in 3D input interaction. We also aimed at creating something novel so, instead of using fiducial or infra-red markers or any kind of active sensors placed in the real setting, to track the physical artifact, we have used color evaluation and motion prediction by computer vision.

Feature detection, tracking, 3D reconstruction and camera calibration are the main issues of our interface, which also includes motion prediction and noise filtering. By bringing together these concepts with the tangible user interface concept, we have come up with ARTIC.

In synthesis, in this paper we present an approach to a tangible interface that allows natural 3D interaction with the virtual world. The paper is organized as follows: in section 2, we provide a background and state-of-the-art overview in the issues of tangible interfaces. In section 3, we present a system overview. Section 4 covers the system development and details the hardware and software of the developed prototype. In section 5, test results and discussion are presented, and finally in section 6 conclusions and future directions of research are given.

2. BACKGROUND AND STATE-OF-THE-ART

Interfaces are systems that have the ability to use physical objects as a form of representation and control of digital information (such as virtual objects). Traditionally with regular Graphical User Interfaces (GUI) we have input devices, such as the mouse, that are used to **control** the digital information, and output devices that enable **representation of** this information. According to Ullmer & Ishii [Ulmer2001], "a central characteristic of Tangible User Interface (TUI) is the seamless **integration** of **representation** and **control**" thus the physical object is tightly coupled to the virtual object for manipulation, visualisation and control.

There are other advantages on using tangible interfaces, for example, by using a graspable object as interface, we can take advantage of its shape, size and position to "in-

crease the functionality and decrease the complexity" Fitzmaurice [Fitzmaurice1996].

According to Kato [Kato2001], TUIs are those in which: "a) each virtual object is registered in a tangible interface; b) the user can interact in real time with virtual objects by manipulating the corresponding tangible object". So in an Augmented Reality context, in order to optimize the interaction between the real and the virtual world it is fundamental to accurately register the virtual objects in the real world. To accomplish this, we need to be able to track physical objects.

The following background examples express the potential of tangible interfaces in a Augmented and Mixed Reality context.

Kato [Kato2001] has proposed a tangible interface for a city planning system on Augmented Reality "A City-Planning System based on Augmented Reality with a Tangible Interface". In this system, the interaction is made using a cup. It is possible to pick up, move or delete a virtual object by manipulating this tangible interface.

In the work of Diniz [Diniz2003] "An Approach to 3D digital design" it is explored the use of simple, inexpensive, non-intrusive devices such as web cameras and small lights (LEDS) to allow a non skilled user to easily start designing. The system tracks the 3D movement of 2 lights in space which are attached to the user fingers, and transforms that movement into ruled surfaces.

In these two examples, tracking is made through computer vision techniques, the next two are also systems that use tangible interface but tracking is not made through the same method.

"Urban Simulation and the Luminous Planning Table: Bridging the Gap between the Digital and the Tangible"[Ben-Joseph2001], is a system proposed by Ben-Joseph and Ishii which aimed at letting the public become more involved in planning and designing physical spaces. The Luminous Planning Table is one of the first prototypes to use a tangible computerized urban design and planning interface.

In "Herding Sheep: Live System Development for Distributed Augmented Reality"[MacWilliams2003], the potential of tangible user interfaces which dynamically visualize, manipulate and control complex operations of many inter-dependent processes has been explored. This project uses the DWARF framework and tracking is made using the optical infrared *DTrack* system from ART GmbH [MacWilliams2003].

3. SYSTEM OVERVIEW

In this work we aimed at creating a tangible interface that is functional for a given application scenario and at the same time providing a new experience of interaction, incorporating the main principles studied by Ullmer & Ishii [Ulmer2001], Fitzmaurice[Fitzmaurice1996] and Kato[Kato2001]. Figure 1 and Figure 2 show the actual shape of ARTIC. It can be described as a 3D handle structure with different axis that can be grabbed as a pen. Its manipulation is very natural and the user can have the immediate perception of what is happening to the manipulated virtual objects when using ARTIC.

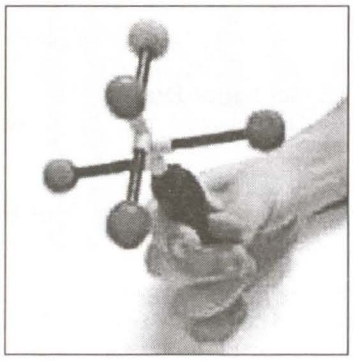

Figure 1 – ARTIC Prototype.

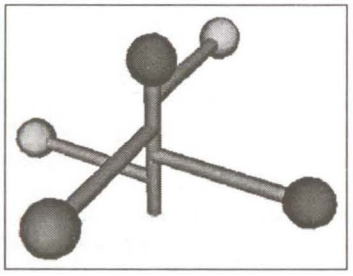

Figure 2 – ARTIC conceptual model.

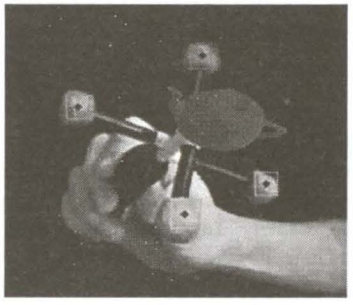

Figure 3 - ARTIC prototype used as 3D input device, allowing for virtual object registration.

Figure 3 illustrates a virtual object registered in the tangible interface. As the user moves the real object, the virtual object flows in the same direction and orientation as the tangible interface.

Our tracking algorithm is based on computer vision techniques, image processing for color evaluation/segmentation and motion prediction. The five colored spheres of the interface structure act as markers that are detected and then tracked in each frame, allowing the tracking of the tangible artifact. There are no other fiducial markers or any kind of sensors used to track the artifact. By knowing the real dimensions of the physical object model and having a minimum of four tracked feature points, we can use the POSIT [OpenCV] algorithm to estimate the ARTIC's pose (position and rotation) in each frame.

Instead of using POSIT for 3D camera calibration, we could have used a homography based algorithm [OpenCV] for a case of a physical artifact with a planar topology, simulating a planar surface, but in this case although obtaining 6DOF, the practical limitations for 3D interaction would be higher.

With 6DOF input devices, interaction becomes simpler, because the user can manipulate virtual objects in 3D space and take advantage of this tri-dimensional world he/she lives in.

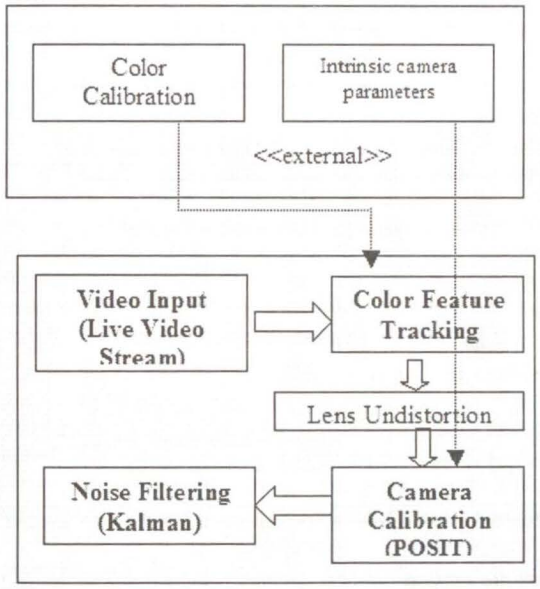

Figure 4 – ARTIC system architecture

Figure 4 illustrates ARTICs system architecture, the external blocks allow us to obtain information about color calibration and intrinsic camera parameters necessary to color feature tracking and virtual camera calibration.

To each acquired image it is applied the color feature tracking algorithm to detect and then track, the colored objects on the image (the sphere markers), in order to track ARTIC. After calculating each colored object's approximate center point, the lens distortion is compensated and the resulting information is used to feed the Camera Calibration algorithm, that is, POSIT. This algorithm needs at least 4 tracked features and a known 3D geometric model of the artifact structure, and returns the extrinsic camera parameters of the virtual camera calibrated with ARTIC, that is, it returns Arctic's position and orientation in the 3D world. Finally Kalman Filtering is used in order to reduce noise in the extrinsic camera parameters time-based laws.

4. SYSTEM DEVELOPMENT

In this section we describe each system module functions and discuss all issues concerning the ARTIC development process.

4.1 ARTIC Tracking

The difficulties that arise in trying to track several colored objects in a series of images, lie in the fact that it is very hard to correctly discriminate object colors when illumination and background changes are not very well known. We had to deal with the color constancy problem since in shadow, all object colors tend to black and similarly, when illumination is too bright, all objects tend to be perceived as white. All following considerations assume small background changes and a constant artificial light environment from the moment of calibration.

4.2 Color Representation

The first step in color processing matters is the selection of a color space for internal color representation, namely RGB or HSV color spaces.

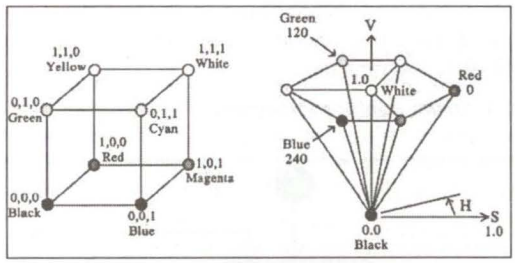

Figure 5 - RGB color cube / HSV color cone

The main advantage in the use of RGB color space is the representation of each color as a quantization of Red, Green and Blue. On the other hand, HSV consists in a transformation of RGB representing colors by a wavelength dependant angle (Hue) and two percentage parameters that reflect the color "purity" (Saturation) and the amount of gray present (Value).

To test each color space behavior and applicability in our case, we have attempted to isolate pre-determined colors by applying a rigid and direct threshold to the image and we found that, although HSV is widely used in computer vision applications and is still a good approximation to human color perception, it was too sensitive to background and inter-object color interference. By evaluating color, represented according to the RGB model, we cease to identify shadowed and highly illuminated areas of the objects but, since at least some portion of each object was detected, the final result proved to be considerably more stable and robust, so RGB was the retained color model.

4.3 Color Calibration

The initial calibration consists in providing the system each object's color reference values. Values are stored in a external file to initialize future uses of the system. The reference values are obtained by asking the user to pinpoint the objects in a specific order. This phase is important, since without correct color calibration we cannot assure coherent results.

4.4 Color Tracking Algorithm

The developed colored object-tracking algorithm can be classified as a region search by adaptive RGB distance comparison.

The algorithm is designed to track multiple colored spheres that, projected in an image frame, result in a group of pixels with similar color values. The center of the colored circular area is assumed to be the center of the area's bounding box.

For each tracked feature color, the algorithm runs independently and can operate in two distinct modes: (1) initial feature detection and (2), feature motion prediction and tracking. Initially the first mode is applied to detect the position of the colored feature on the image. Having successfully located the feature position, the second mode is triggered. When feature motion prediction and tracking mode fails, the algorithm reinitializes returning to the first operating mode.

4.4.1 Initial Feature Detection

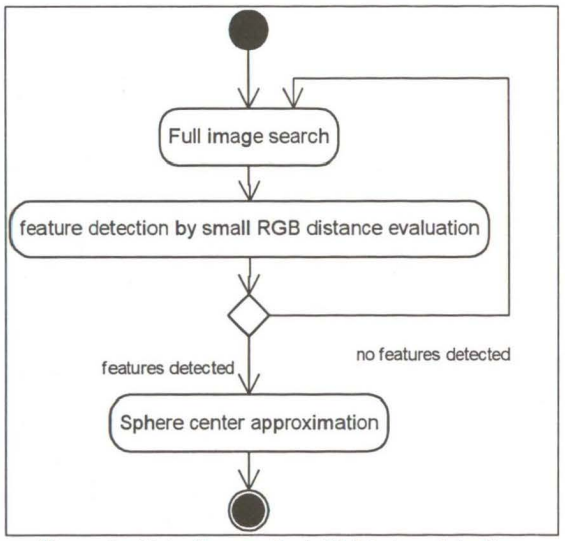

Figure 6 – State diagram for initial feature detection.

When operating in initial feature detection mode, the algorithm has no information about where the features may be located, so every pixel in the image is evaluated in order to obtain the feature's position.

To understand the actual pixel evaluation method we need to bear in mind the RGB color cube, once every input pixel is evaluated in terms of the distance between its color values and the feature reference color. Basically this reduces to selecting pixels with color values located in a sphere inside the RGB cube, centered on the object's reference color with radius equal to a given comparison distance. The actual distance measuring is done by pre-calculating all distances and storing them in a lookup table to reduce real-time processing.

To avoid inter-color interference, it is considered a small RGB distance to the original calibrated reference color value. Assuming a small RGB distance for pixel evaluation also means that only a portion of the colored sphere will be identified since darker areas will correspond to pixels located in a RGB distance larger than allowed.

4.4.2 Feature motion prediction and tracking

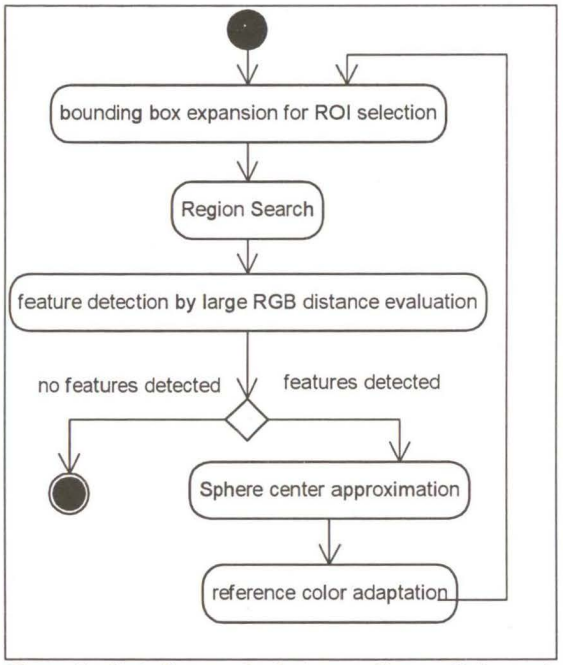

Figure 7 – State diagram for feature motion prediction and tracking.

Feature motion prediction and tracking mode is applied only when the algorithm has information about a feature's recent position, namely when initial feature detection is successfully completed, or when the algorithm is operating repeatedly in this mode.

Knowing a feature's position in the last frame, there is no need to search every pixel of the image, so pixel evaluation is done only around it's last known position establishing a region of interest where the search is made.

This method dramatically reduces processing load, resulting in a higher frame rate.

The pixel evaluation method is the same as used in initial feature detection mode, with the exception that a larger RGB distance is used for comparison around an adapted reference pixel value.

We can afford to consider a larger distance because it is unlikely that pixels from more than one colored sphere are located in the search area. The larger comparison RGB distance allows the algorithm to identify a larger area of the sphere and consequently provide a better approximation of its center.

The reference color values information for each tracked object is also altered at run-time, to ensure that color variations resulting from the structure's position are taken in consideration. The reference values adaptation is done through a weighed average between the original

reference value and all values inside the considered RGB sphere. The original reference value is stored at color calibration time and is never deleted so we can use it when tracking is lost.

4.4.3 Alternative Tracking Algorithms

An alternative to the above color tracking method is the use of an image histogram's back-projection as used in the color tracker example from the *CamShift* Example provided in *OpenCV*'s demo applications [OpenCV]. This technique calculates a histogram over the hue plane of an image containing only the object in question so the maximum will correspond to the object's color. This histogram is used to replace each pixel value with its corresponding probability. The result is a light area corresponding to the object's position over a black or very dark background. Although very robust and fairly fast for one object, the processing load imposed to track several objects considerably slowed down our system due to the repeated histogram and back-projection calculations. Another disadvantage was that, to ensure an accurate tracking, the algorithm had to be correctly calibrated, meaning that the final user was required to possess some knowledge of histogram parameters and behavior.

Further improvements to the algorithm involved, tracking the object through its shape, exploring the fact that if each tracked object is a sphere then they would always appear in the image as circles. Testing was performed on circle detection first using a *Canny edge detector*[OpenCV] and secondly applying color threshold followed by a circle mask comparison and connected component search. Both methods proved to be very unstable (also due to the relatively small size of the feature objects in the image) and too sensitive to natural image noise while demanding too much processing time, so no improvements were included in the final prototype.

4.5 Camera Calibration - POSIT

By knowing the real dimensions of the physical object model and having a minimum of four tracked feature points, we can use the POSIT [OpenCV] algorithm to estimate the ARTIC's pose (position and rotation) in each frame.

This information (ARTIC's pose) finally allow us to use ARTIC as a 3D input tangible device.

POSIT algorithm finds the pose of an object from a single image. Necessary conditions for the algorithm to work are: the extraction of at least four non-coplanar points as features in the image domain to allow matching of the extracted features with the corresponding known object model points. It combines two algorithms. The former, called POS (Pose from Orthography and Scaling) [DeMenthon1995], approximates the perspective projection with a Scaled Orthographic Projection (SOP) (Figure 8) and finds the rotation matrix and the translation vector of the object in the image, in the camera reference frame, by solving a linear system. The latter is an iterative method that applies POS to the approximate pose found in the previous step in order to compute better SOP of the feature points and, for this reason, is called POSIT (POS with Iterations) [DeMenthon1995]. Many improvements since the proposed algorithm have been published making it a powerful and fast tool for pose estimation.

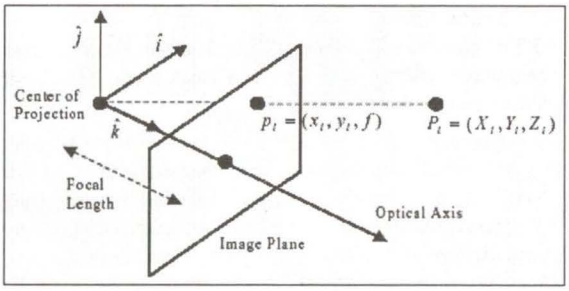

Figure 8 – Scaled Orthographic Projection camera model assumed in POSIT

ARTIC uses OpenCV's implementation of the POSIT algorithm [OpenCV]. This implementation receives as input parameters the coordinates, in the image domain, of each colored feature, the corresponding physical model's geometry and topology, the focal length, the precision and the number of iterations. POSIT algorithm returns a translation vector and a rotation matrix (extrinsic camera parameters) corresponding to the artifact pose in the current frame. The 2D locations of each feature are given by our color tracking algorithm and the physical model information (geometry and topology) is built during initialization and known "a priori". The number of iterations and estimation precision are dimensioned to provide a fairly good estimation without overloading real-time processing. The focal length (used only as a measure reference in POSIT) and the remaining camera intrinsic parameters are pre-calculated using *OpenCV's* [OpenCV] calibration tools.

The final prototype tracks five colored objects, for a higher stability and to reduce POSIT estimation errors. Since ARTIC's colored objects rotation and translation information is relative to a given object position and orientation of the model, some additional computing is necessary. This implies that the order in which we supply the features position to the algorithm has to be the same as the one used to define the colored objects in the physical artifact topology. Since POSIT needs at least four non-coplanar points, we define several versions of the artifact's topology, considering all possible combinations of extracted features, including a combination that uses all five and, at run-time, decide which one should be used by POSIT, depending on which features were currently successfully identified. By convention we define all colored object's pose information relative to the top-most colored object, and if this specific object was not detected (assuming all other four were) we use one of the others as reference and modify the algorithm's returned pose information to define it in relation to the top-most object.

From our experiments we found that, in order to work consistently, the colored spheres should be placed in fairly positions distant on each axis since this improves the estimation even when only four features are used.

We have noticed a limitation of OpenCV's implementation of the POSIT algorithm: it returns incorrect results when the object's pose reflects a rotation about 90° over one of the axis.

4.6 Noise Filter

ARTIC uses two instances of the Kalman filter to deal separately with the rotations and translations (extrinsic camera parameters).

To empirically test the result's validity, we have tried to register virtual objects onto our prototype and what we found out, was that some kind of filtering was required for camera stabilization. Our first approach was the implementation of a simple low-pass filter by averaging consecutive feature points coordinates. Although fairly effective, this method introduced a large amount of error since distorting the 2D positions led to uncertain results when 3D reconstruction was triggered. The solution was to use a Kalman Filter to filter extrinsic camera parameters. The Kalman filter consists in a set of mathematical operations used recursively to estimate the state of a generic process in a way that minimizes the mean of the squared error. It is very efficient, fast and robust allowing the use of previous or future states. Another interesting and useful characteristic is that it considers time as an influencing variable. An implementation of Kalman filter is available in OpenCV [OpenCV] and is used in our project. Kalman filtering is performed over the most recent 3D information obtained in order to use a Kalman updated prediction instead of the original results. The filtering effect can be controlled by altering the termination parameters and reference covariance error. These parameters deeply influence results, since a smaller error reference produces softer results while increasing jitter. On the other hand, the error reference has to be small enough to compensate small estimation error without generating a perceivable delay.

4.7 Material Selection and Topology Design

To build the physical structure that supports the colored objects we have to consider that both the chosen material for the objects and their topology or position in the structure should boost the system's potential while minimizing the effects of the system's limitations.

The structure is built with several connectable plastic pieces that make the model very light and easy to handle.

The first approach in colored object's material selection consisted in using lit up multicolored leds, based in research results of a project also developed by a member of our research team, using two leds to draw 3D objects by detecting their glow in low light environments, described in the work of Diniz [Diniz2003]. Since the leds have their own light they are immune to shadows in what concerns tracking. As referenced earlier, each object has to be univocally identified but we were limited to the existing colors available in the led market. A major defect in led usage, is that the camera captures apparently different colors as similar, for example yellow light from the led is captured as orange and orange as red. After several tests the selected led colors were red, yellow, blue and green since they formed the least interfering combination of possible colors. The disadvantages consisted in the camera capture of light diffusion patterns and the shape of the leds that directly influenced the structure's topology, and consequently limited freedom of movement. For the leds to be illuminated they had to be connected to a nine volt battery which besides adding weight to the structure proved to be most uncomfortable because it had to be placed near the user's grip zone.

Ironically the major advantage in led usage, namely the fact that they possess their own light, turned to be the reason why we decided to use other materials. Each led didn't emit only one color, which made the camera capture several wavelengths around the target color. For example, while capturing the green led the result was white a zone in the vicinity of the central and brightest point of the led, surrounded by yellow, orange, blue tones and finally the green pixels around the led's edges, making it impossible to eliminate inter-object interference and correctly identify each object univocally. This phenomenon was noted on every led tested.

At this point we were certain that the selected material had to be opaque, even if it didn't possess their own brightness. It would also have to possess one clearly defined non-reflective solid color. Our targets were colors that stayed as far apart as possible in the RGB color cube representation, namely its corners representing red, blue, green, yellow, magenta or cyan. The selected material was plasticine since it complied with the earlier mentioned characteristics, plus it was moldable, making it easy to mount it in the structure, and it is widely commercialized in different color allowing better color selection.

The selected plasticine blocks were molded as approximately 20mm diameter spheres. The artifact's size was defined considering that the structure would be most of times in one arm's length distance away from the camera so it had to be small enough to be portable, but large enough to always be identified on the image. Each object's solid color helped to stabilize the color tracking algorithm leaving little room for interference as the whole area of each object is identified producing a much better approximation to each sphere center, improving the 3D reconstruction results as well. The disadvantage is that we were now sensitive to illumination direction since we are no longer able to track the objects if they are placed in a shadowed area. Still we found this to be a better solution to our problems considering only vision-based color tracking methods.

4.8 Topology Evolution

The design of the physical structure evolved as the used materials changed over time and as algorithms were improved, making software development and structure design evolve on proportional rhythms. We produced several different prototype topologies that led to the final structure topology.

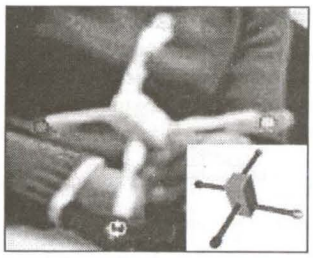

Figure 9 - ARTIC Prototype 1.0

The first prototype (1.0, Figure 9) was built only to give us an idea on what could we expect to achieve and the major difficulties we would encounter on developing ARTIC. It consisted in four leds placed in different x, y and z coordinates. The leds were supported by four 70mm long plastic straws attached to a polystyrene cube. What we found was that, due to their shape, the only way to capture each led's colors was to force them to always be in a frontal position relative to the camera. The leds also proved to be too small, making it very hard to identify them at longer distances.

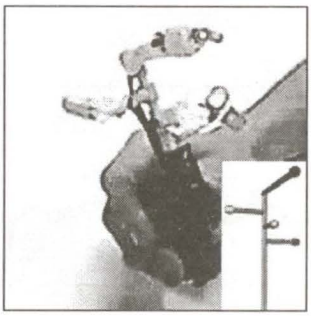

Figure 10 - ARTIC Prototype 1.1

Now that we knew how the difference in axis coordinates influenced the 3D reconstruction precision we were more careful in choosing where to place each object. To compensate some color's difficulties in tracking, we added more leds of each color and grouped them so the colored area would be larger for each object (Prototype 1.1, Figure 10). This fact introduced errors in reconstruction since the identified center of the led group didn't always correspond to the real center location plus, since the colored area grew, we got an ever larger variety of interfering colors due to the camera's capture of colored glow. At this point we started to use the plastic connectors to build the structure and, because they are also colored, they were covered with tape to avoid interferences. The structure was mounted on a handle and now we were wondering how to fit the 9V battery in the structure without making it in uncomfortable. To make sure that all leds were in a frontal position we had to limit user's freedom of movement, being able to rotate only to positions where all four leds were visible (approximately 180° about *x*, *y* and *z* axis, Figure 10).

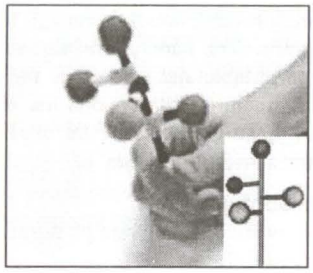

Figure 11- ARTIC Prototype 2.0

Figure 11 shows our first experiment using plasticine, and we tried to apply the same ideas from its predecessor. We found that each object had to be further apart because the tracked objects are now 20mm diameter spheres. Since tracking of each colored object improved, smoother results were obtained in reconstruction although user's freedom of movement didn't yet allow full 360° rotation for every axis.

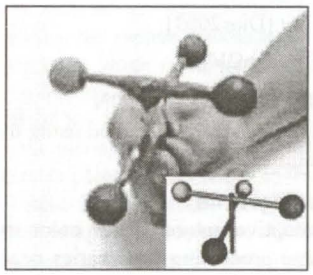

Figure 12 - ARTIC Prototype 2.1

In Figure 12 we experimented and designed a new combination of coordinates for sphere placement, leading us to the initial idea of spreading the objects around in all three axis, in such a way that it would be difficult to occlude one of the spheres. Disadvantages in this topology were based on the lost of accuracy, since two spheres were placed in the same y coordinate and we still had the limitation that all four colored objects had to be identified so the system would work.

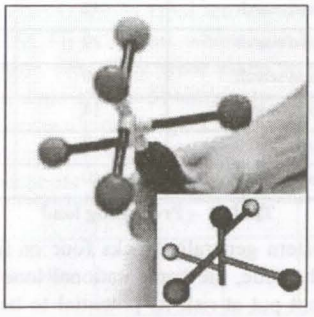

Figure 13 - ARTIC Prototype 2.2

Figure 13 is the actual ARTIC physical structure topology. It represents a serious improvement from the earlier versions mostly thanks to the inclusion of the fifth colored object. This conduced to an improvement in precision since more features were used in estimation, and we could now afford to lose track of one object without completely losing 3D information since normally at least four

four objects are identified which is enough for 3D virtual camera calibration. The handle was also revised to become more comfortable and ergonomic for easier usage and, since the structure that supports the objects is detachable, other handle types may be used to facilitate usage and better satisfy user needs.

4.9 System Configuration

The typical hardware and software platforms required by our system, are as follows:

- Hardware:
  Intel Pentium III, 1GHz, 256 KB RAM
  NVIDIA GeForce2 MX/MX 400
  Web cam Creative NX
- Software:
  Windows 2000 Professional
  OpenGL [OpenGL]
  MX Toolkit [Dias2003]
  OpenCV [OpenCV]

5. RESULTS AND DISCUTION

The following results were obtained using the above system configuration.

5.1 Processing Time

Due to the adaptive nature of our color tracking algorithm the frame processing time varies according to the number of colors it is trying to track, in a small region or over the full image frame. The most efficient state is when all colors are searched in previously selected areas, the less efficient state occurs in the first input frames when all colors are in full image search mode. Table 1 discriminates results for the above described system's configuration.

	Output frame rate (fps)	Single frame processing time (ms)
5 colors in region search	42	24
4 colors in region search	27	37
3 colors in region search	19	53
2 colors in region search	14	71
1 colors in region search	12	83
0 colors in region search	10	100

Table 1 – Processing load

Since the system generally tracks four or five colors in region search mode, the computational load ARTIC imposes does not put at risk its potential to be included as an input device in augmented reality applications.

5.2 Interface Stability

One of our concerns was to provide consistent and continuous 3D information as accurate as possible. The following graph information was obtained holding the structure in a random still position over several consecutive frames while recording the returned values.

In the following representations the unfiltered results are represented by the thin line and the Kalman filtered information is represented by the thick line. The vertical scale represents the system's output values in *mm* and the horizontal scale represents the number of frames.

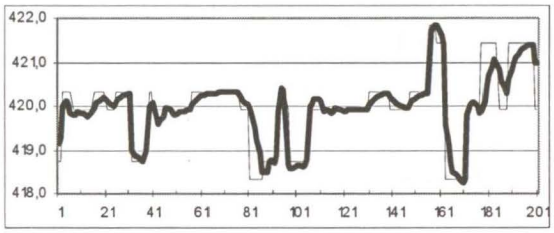

Figure 14 –Still translation values over x axis

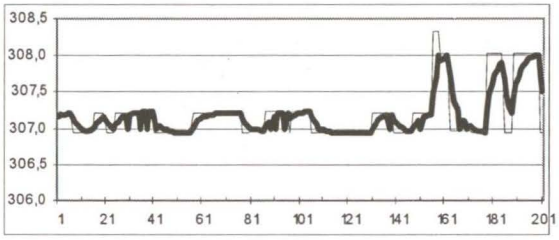

Figure 15 – Still translation values over y axis

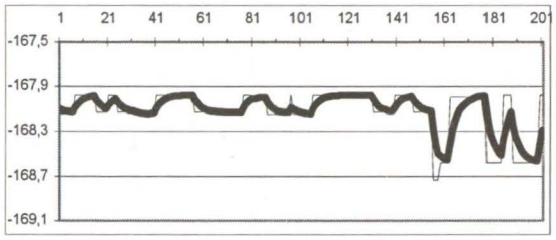

Figure 16 – Still translation values over z axis

From the above representations we can observe small variations that occur even with no movement. Values vary in a maximum of 2mm around the reference value. We can observe the effect of Kalman filtering making smooth transitions to minimize the effect of input noise.

As a result of Kalman filtering, some jitter is introduced to the system and can be identified in the above representations. The following representations are the result of performing the same test with the structure in motion.

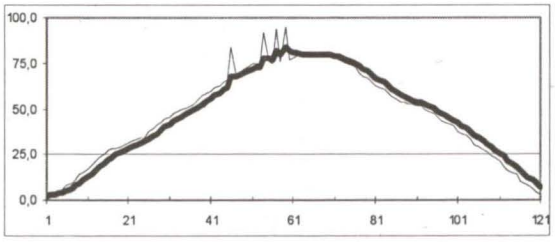

Figure 17 – moving translation values over x axis

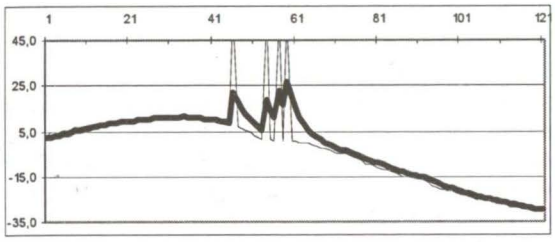

Figure 18 – moving translation values over y axis

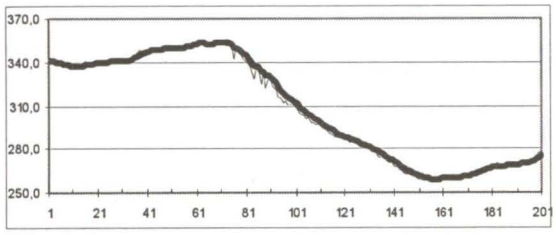

Figure 19 – moving translation values over z axis

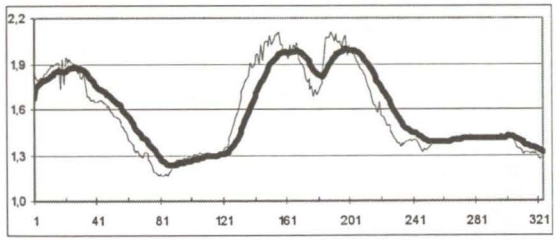

Figure 20 – moving rotation values over x axis

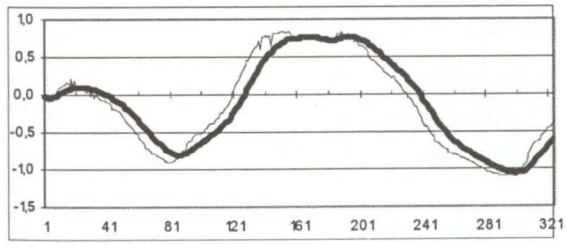

Figure 21 – moving rotation values over y axis

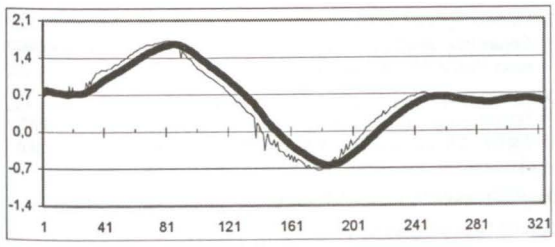

Figure 22 – moving rotation values over z axis

Figures 17 to 22 illustrate the Kalman effect when the structure is in motion, as expected, some jitter can be observed as well as the elimination of undesirable noise peaks for both translation and rotation values.

5.3 Camera Calibration Precision

Precision test consisted in comparing the system's output when known transformations are applied to the structure. To control these transformations we feed the system with images of a moving ARTIC virtual (VRML) model extracted from the OpenGL frame buffer. Recording both the applied transformations and the system's output on each frame, we are able to estimate the average error for translations and rotations over each axis.

The structure's virtual model is rendered assuming the default camera parameters of ARToolkit, which implies that the same camera parameters are considered in POSIT configuration. The test was performed to approximately 900 consecutive frames, at a frame rate of 30fps over a period of 30 seconds.

. Average error in translations over x axis	0,21 mm
Average error in translations over y axis	2,93 mm
Average error in translations over z axis	1.268 mm
Average error in rotations over x axis	5,72°
Average error in rotations over y axis	24°
Average error in rotations over z axis	28°

Table 2 – Average error in transformations

As presented above, the system returns translation values with higher precision when compared to rotations. We think this fact is directly related to limitations of the used implementation of the POSIT algorithm since, as stated earlier, it returns inconsistent rotation values when the tracked structure is placed in an angle in the vicinity of 90° over every axis. Because major contributions to rotation errors occur in specific positions, the average error distribution consists in a group of error peaks that damage rotation precision. We believe that once the 3D camera calibration errors are eliminated from our system, the rotation results will be as accurate as the translation output. The delay jitter introduced by Kalman filtering also introduces has a small contribution to the mean error on each frame since the time of measurement by our algorithm is synchronised with the transformations in the virtual model, that serve as a reference.

5.4 Usability Testing

In order to obtain an impartial view on the prototype's usability we used opinions from ten unpaid student volunteers from ISCTE in Lisbon, whom ages were between 19 and 23 years old. They were asked to perform and evaluate (in a scale of 1 to 5) simple operations of rotating and translating a virtual object registered on ARTIC's physical structure. Each experiment lasted about 5 minutes. Users' response is depicted in the following table.

	1	2	3	4	5
Correspondence between real and virtual movement	0%	0%	30%	60%	10%
Natural/Simple usage	0%	0%	0%	70%	30%
Non-intrusion and handling comfort	0%	0%	20%	60%	20%
Acceptance if used in applications	0%	0%	20%	50%	30%

Table 3 – User evaluation results

All users rapidly got used to handling the prototype and performing the requested operations. User evaluation was

clearly positive as the majority of users rated 4 in all categories. They found the system's response to be satisfactory while examining virtual objects since it was easy the view all parts of the object from different perspectives. They noted the incorrect results provided by OpenCV's implementation of the POSIT because, while freely moving the structure, it was easy to place ARTIC in a position with 90° on one of the axis, making the errors noticeable in virtual object registration.

As we can observe from the results, none of the users rated 1 (bad) or 2 (poor) in any of the performed tasks. In the first task 30% of the users felt that the correspondence between the real and the virtual movement was satisfactory (3) which can be explained by the use of Kalman filter that smoothes the movement although introducing some jitter. In the same question the majority (60%) thought that the system had a good (4) performance and 10% rated very good (5). We can also verify that 100% of the users felt that ARTIC has a good or very good level of natural usage. Non-intrusion and handling comfort as well as acceptance in applications had similar results, 20% felt that it was satisfactory and 80% respond that it was good/ very good.

6. CONCLUSIONS AND FUTURE DIRECTIONS

The system presented in this paper describes a novel tangible interface that enables 3D user interaction and that can be applied in the context of AR and MR. The interface explores the use of a physical, attractive and simple artifact to enrich the HCI process. We were able to build a low-cost system using only common materials (such as plastic and plasticine) and that requires just a webcam, which were one of our primary goals. We have demonstrated that is possible to conceive tangible interfaces providing 3D input, with no requirements of active sensors and completely portable. This interface is a low cost alternative to marker-based systems, either using vision and color or black and white fiducial markers, or infra-red cameras and infra-red markers, although ARTIC is limited to unprepared real settings backgrounds, but where the colors of the artifact are not present.

From the usability test we can conclude that we have also achieved another important goal such as the simplicity of the tangible interface and natural, non-intrusive and comfort handling, though we are studying other forms of handling and usage of ARTIC on concrete applications. In fact, all the early results show that ARTIC provides an easy way to interact with virtual objects, making it a potential tool for use in AR/MR applications

The processing time test showed that our adaptive tracking color algorithm is efficient and it does not require long processing time.

Other implementations of the POSIT algorithm will be tested to try to solve current errors in specific pose angles.

Several future directions can be thought for this kind of system, namely the deployment of applications that uses ARTIC as a 3D input device, performing tasks such as sketching, picking, examining, zooming, panning, etc. Envisaged applications are, for example architecture or even interior design in AR/MR settings. We are already studying a virtual picking button in ARTIC. Depending on the requirements of the applications it could be also possible to have more than one ARTIC tangible interface in the interaction process, which would imply tracking multiple interfaces, using distributed system architecture and multiple cameras.

7. ACKNOWLEDGEMENTS

The authors would like to thank Nancy Diniz from ISCTE, Portugal, for her initial contributions and discussions about tangible interfaces for conceptual design.

8. REFERENCES

[Ulmer2001] Ullmer, B., Ishii, H. "Emerging Frameworks for Tangible User Interfaces." in *Human-Computer Interaction in the New Millenium* John M. Carroll, ed.; © Addison-Wesley, August 2001, pp. 579-601.

[Fitzmaurice1996] Fitzmaurice, G., "Graspable User Interfaces" Ph.D. Thesis, University of Toronto,1996. http://www.dgp.toronto.edu/%7EgI/papers/PhD%20-%20Graspable%20UIs/Thesis.gf.html

[Kato2001] Kato, H., Billinghurst, M., Poupyrev, I., "Tangible Augmented Reality", in *Augmented Reality: the Interface is Everywhere*, SIGGRAPH 2001 Course Notes 27, 2001

[Ben-Joseph2001] Ben-Joseph, E., Ishii, H., Underkoffler, J., Piper, B., and Yeung, L. "Urban Simulation and the Luminous Planning Table: Bridging the Gap between the Digital and the Tangible", in Journal of Planning in Education and Research, Volume 21, pp.195-202, 2001.

[Diniz2003] Diniz, N., "AN APPROACH TO 3D DIGITAL DESIGN Free Hand Form Generation", DCC'04 MIT 19-21 July 2004.

[MacWilliams2003] MacWilliams, A., Sandor, C., Wagner, M., Bauer, M., Klinker,G., Bruegge, B., "Herding Sheep: Live. System Development for Distributed Augmented Reality", in Proc of ISMAR'03, pp 123-132, 2003.

[DeMenthon1995] DeMenthon, D. and L.S. Davis, "Model-Based Object Pose in 25 Lines of Code", in International Journal of Computer Vision, 15, pp. 123-141, June 1995.

[Dias2003] Dias, J., M., S., Jorge, J., Carvalho, J., Santos, P., Luzio, J., "Developing and Authoring Mixed Reality with MX Toolkit", ART03, The Second IEEE International Augmented Reality Toolkit Workshop, Tokio, Japan, 6th October 2003.

[OpenCV] OpenCV, www.intel.com/research/mrl/research/opencv/

[OpenGL] OpenGL, www.opengl.org