The 8th ACM/EG Expressive Symposium
C. Kaplan, A. Forbes, and S. DiVerdi (Editors)

Defining Hatching in Art

G. Philbrick[†] and C. S. Kaplan[‡]

University of Waterloo, Canada

Abstract

We define hatching—a drawing technique—as rigorously as possible. A pure mathematical formulation or even a binary this-or-that definition is unreachable, but useful insights come from driving as close as we can. First we explain hatching's purposes. Then we define hatching as the use of patches: groups of roughly parallel curves that form flexible, simple patterns. After elaborating on this definition's parts, we briefly treat considerations for research in expressive rendering.

CCS Concepts

• ***Computing methodologies*** → ***Non-photorealistic rendering; Image processing;*** *Texturing;*

1. Introduction

Hatching is a drawing technique that uses patterns of roughly parallel curves to achieve multiple ends, the most traditional of which are low-level, depictive goals like creating gray levels and showing form and texture. Hatching also fills higher-level symbolic and compositional roles.

Figure 1: Hatpin Girl, *Charles Dana Gibson, pen and ink*

Figure 1 shows hatching's depictive versatility: marks cooperate to establish lights and darks, show the curvature of forms, and suggest texture. The paradigm in this particular image is conservative and pre-modern; realistic depiction is Gibson's preoccupation here, and he does not deliberately employ hatching for symbolic or purely aesthetic purposes. Yet the hatching marks do have a beauty of their own, an appeal separate from the illustrative purposes they happen to serve.

Hatching is part of *drawing*, which we tentatively define as 2D art whose primitives are curves—"lines" to some artists. The tool used to make them does not matter, whether it be a pencil, pen, paint brush, engraving burin, or while-loop, so long as the resulting image contains shapes that read as curves. Not all drawing marks are hatching, so "hatching" identifies a group of marks, not necessarily an entire image.

The purpose of this paper is to give hatching a technical definition useful for research. Before arriving at this definition, we elaborate on hatching's roles, familiarity with which will make the definition more sensible. The definition is a two-purpose tool, meant both for generating and identifying hatching. Researchers can use the definition as a guide for generating hatching in any context, whether in rendering 3D geometry, artistically "filtering" existing images, or executing user requests in a drawing program. The definition will be even more useful for identifying and interpreting existing hatching, since taxonomical issues are more salient here.

After presenting our definition of hatching, we address issues with generating and analyzing hatching in research on expressive rendering.

[†] gphilbri@uwaterloo.ca

[‡] csk@uwaterloo.ca

DOI: 10.2312/exp.20191082

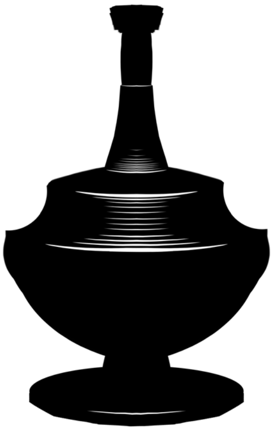


Figure 2: *White-on-black hatching. Unless otherwise noted, all drawings are by the first author.*

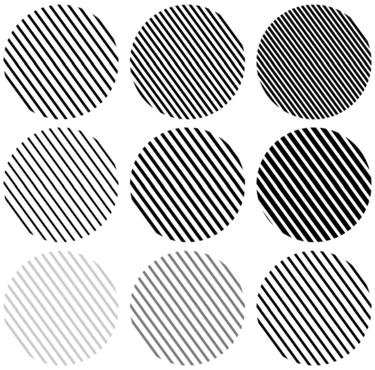


Figure 3: *To create different "optical grays" with a single layer of hatching, vary marks' density (top row), their widths (middle row), or the value of the ink used to draw them (bottom row).*

2. Purposes

It is apparent that hatching originated from necessity. Some artistic media require artists to rely exclusively on marks made with a sharp instrument—points and curves in effect. Hatching is one response to this limitation: it turns the lowly curve into an all-purpose illustrative primitive (similarly, stippling [MARI17] turns the point into an all-purpose primitive). Printmaking media often involve the kinds of constraints that make hatching useful, which is why hatching has a long history as a halftoning [LA08] technique.

Over time, hatching has come to serve the three basic depictive purposes: value, form, and texture. It also has uses as a symbolic device and as a compositional tool. Finally, hatching is intrinsically appealing, even when not clearly serving one of the five foregoing purposes. This appeal is a sixth purpose in itself.

Value (or "tone" [Loo12]) means brightness or gray level. Conventionally, hatching involves marks that are pure black, such as when carving a groove into a woodblock or stroking a page with a quill pen. Marks do not blend physically for the most part, so creating values other than pure black or pure white requires marks to blend *visually*, making hatching a de facto halftoning technique like stippling. Hatching marks can be in any tone or color, of course (Figure 2).

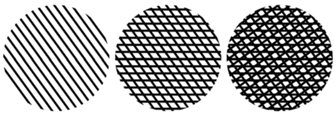


Figure 4: *Crosshatching is a way to produce different optical grays using multiple layers of hatching.*

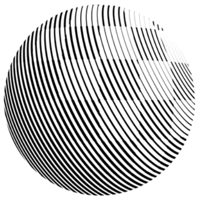


Figure 5: *Value does more than just show form. Here, gray levels do establish the sphere's form through light and shadow, but they also establish its (monochromatic) color pattern.*

Hatching does not directly produce a desired gray; marks act together to create the impression of one—an "optical gray" [Ens03, p. 151]. Artists create different optical grays by altering curves' density, their widths, or the value of the ink used to draw them (Figure 3). They may also use multiple layers of hatching marks moving in different directions to produce *crosshatching*, as in Figure 4. (For some artists, "hatching" and "crosshatching" are mutually exclusive terms, the first applying only to a single layer of hatching marks. In contrast, we treat crosshatching as a specific technique under the hatching umbrella.)

Any drawing marks that establish value can also establish form (and more—see Figure 5). This is because value encompasses *shading*: showing which sides of objects are in light or in shadow, along with cast shadows, core shadows [Gur10, p. 46], and highlights. Yet there is a second mechanism for portraying form which is unique to hatching and completely unrelated to value: if the curves wrap around an imagined 3D surface, the eye will perceive depth, even without lighting cues (Figure 6). Imagine starting out with 3D curves based in the scene, specifically isocurves within one of the scene's surfaces. When two of these isocurves are projected to the image plane to produce hatching marks, the varying distance

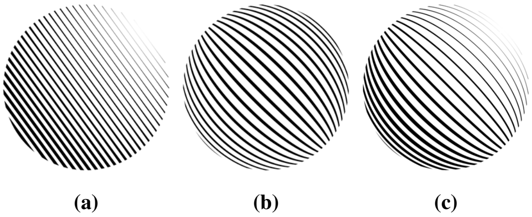


Figure 6: *Hatching can show form through value variation (a), bending marks and varying their placement (b), or both (c).*

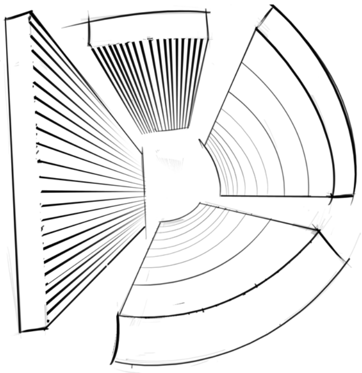


Figure 7: *Varying hatching marks' spacing or having marks converge can produce the effect of perspective.*

Figure 8: *Hatching can illustrate texture on the surface of macroscopic forms.*

between the marks will show the surface's curvature and the foreshortening caused by perspective (Figure 7). Art literature sometimes calls such hatching marks *cross-contours* [Ens03, p. 130]. Principal curvature is often a good choice for guiding the surface curves that get projected into cross-contours [GIHL00]. There are other, more specific tricks for conveying perspective when hatching cylindrical objects [MSVF09, VSE*06].

Before explaining how hatching depicts texture, we must disambiguate "texture." In our writing, this term denotes form at small scale: *microforms* in a three-dimensional scene, or the illusion of these microforms created in the mind of the viewer by hatching. It does not have the meaning it often does in 3D graphics contexts, where it refers to a color pattern on a perfectly smooth surface, like in Figure 5. Nor does it refer to the interplay of 2D marks within the image plane—this paper uses "pattern" for that purpose.

Figure 9: *Hatching can illustrate texture that emerges from numerous separate microforms, like hairs or blades of grass.*

There are two kinds of microforms. In the more intuitive case, a macroscopic object has minor surface details which are forms in their own right, like the ridges on a shell or on the cut surface of a piece of wood (Figure 8). Less intuitively, a group of separate microforms can appear as a single form: 100,000 individual hairs simplify into a hairdo; millions of blades of grass coalesce into a collection of tussocks or a giant prism (Figure 9). The kinds of texture ideal for hatching are those whose microforms are grooves or strands.

A single hatching mark does not strictly correspond to a single microform. When working from life or from imagination, drawing every hair, feather, or blade of grass is prohibitively difficult and often produces an unappealing result anyway. Thus, hatching is a shorthand for texture, with varying degrees of abbreviation. Take the character Woodstock from Schulz's *Peanuts*, whose head of plumage can be represented by as few as three marks (this is hatching as visual abstraction [Dod18]). A related kind of visual abbreviation is to detail the texture only on certain parts of a form, letting the mind fill in the blanks; Winkenbach and Salesin [WS94] and Grabli et al. [GDS04] call this *indication*. Figure 10 displays both types of abbreviation.

There is a principle here: if a group of hatching marks represents a texture to the viewer, they will keep representing more or less the same texture even if some marks are added or taken away. Later we generalize this principle and claim that flexibility is an essential aspect of hatching. Whatever pictorial role a group of hatching marks currently fills, it will keep filling that role subject to modification of density, neatness, direction, and so on.

With value, form, and texture, hatching already has the power to depict scenes as an eye or camera would take them in. Yet hatching can do more than by-the-book rendering.

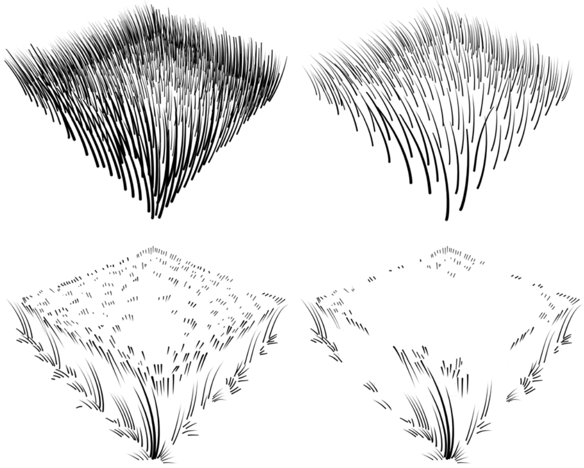
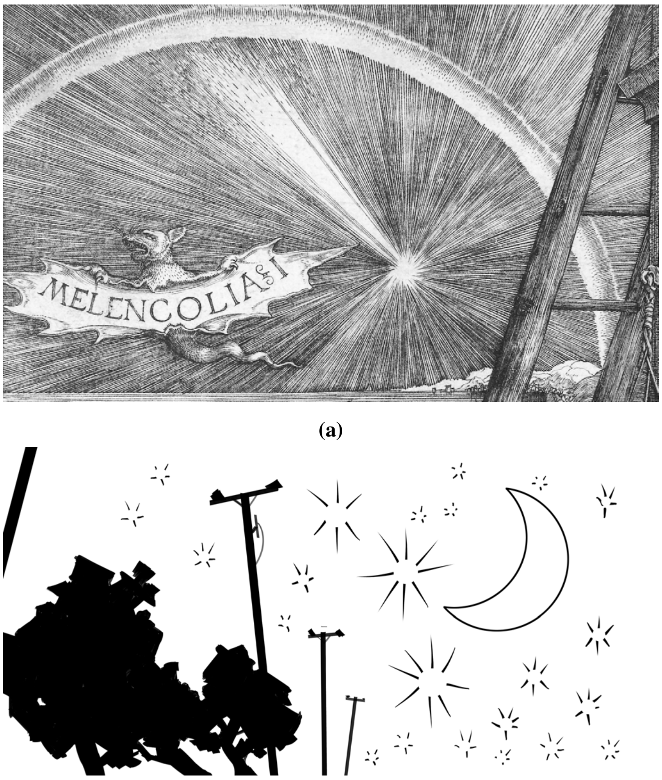


Figure 10: *Hatching is a shorthand for texture. In this example, the number of marks need not match the number of actual or presumable blades of grass. In the first row, different numbers of marks express the same idea: a patch of grass. Similarly, the artist can leave out some areas of texture, letting the viewer interpolate mentally (second row).*



(a)

(b)

Figure 11: *Detail from* Melancholia*, Albrecht Dürer, 1514, copper plate (a); night sky with stars as radiating hatching marks (b)*

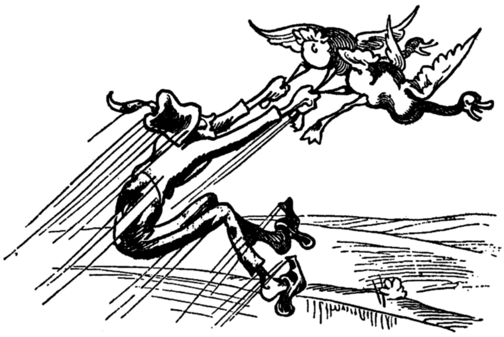


Figure 12: *Some hatching marks are purely symbolic, like action lines. Illustration from* Max und Moritz*, Wilhelm Busch, 1865, wood engraving.*

Figure 13: *Hatching has compositional uses like energizing the scene and leading the eye.* Trajectory to Taurus*, Virgil Finlay, pen and ink. Copyright Lail M. Finlay, used with permission.*

Figure 11 shows two examples of how clusters of radiating lines—a kind of hatching—can read as light sources. In the Dürer detail (top), the effect could be called simply depictive because the eye sees streaks of white against a dark background, matching the experience of looking at a bright light. However, in the stylized night sky (bottom), the groups of radiating lines are black against white, not what a camera or an eye would record. The artist is using symbols rather than depicting veridically. In comics and similar art, hatching takes on highly semiotic forms, like action lines for showing motion (Figure 12).

The use of symbols is one of the paths leading away from basic rendering. Another is composition, which operates at a higher level than merely showing what a scene looks like. It is not just about putting information in the image plane, but also involves guiding the eye to the most important information, making different parts of a picture work together to create beauty, and inspiring emotion. Figure 13 shows hatching applied with a compositional purpose. Finlay uses radiating lines to depict a light source like Dürer does, but in a creative, multipurpose way. The radiation covers the whole

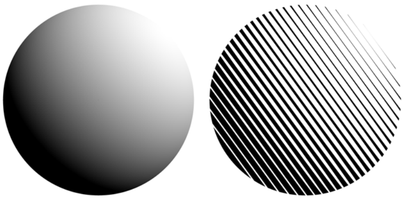


Figure 14: *The hatched scene at right is functionally just an imperfect optical-gray approximation of the ideal scene at left, yet we argue that the hatching is more beautiful.*

scene, not just indicating light but guiding the eye and giving energy to the composition.

Hatching might even abandon its connection to a scene entirely and become pure ornamentation. According to Aldous Huxley, drapery (folded cloth) in art is an end in itself without regard to the scene around it [Hux52]. Hatching has the same context-transcending charm. Its beauty might just derive from the viewer's unconscious appreciation of a correctly rendered scene, or it could be deeper than that. Figure 14 illustrates the question. The bending of hatching marks to show surface curvature is deliberately omitted here. The straight hatching marks at right do nothing but produce optical grays. Yet even though the hatching is, in a way, just an imperfect rendering of the "true" scene at left, the hatching is more interesting, suggesting that hatching's appeal is at least partly independent of its function within a scene.

3. Definition

Our definition of hatching implicitly encompasses all the purposes just described and accommodates both human- and computer-generated hatching:

> Hatching is the use of one or more patches, where a patch is a group of roughly parallel curves that form a flexible, simple pattern.

The terms *curve*, *patch*, *flexible*, and *simple* need to be elaborated upon. Be aware of the vein of subjectivity running through the elaboration. This subjectivity is not a problem when creating hatching; the just-presented definition is almost sufficient for that task without any further explanation. It is in classifying hatching—in analyzing a group of marks and answering the question, "Is it hatching?"—where subjectivity is crucial and unavoidable. There will inevitably be cases where the answer is neither yes or no.

3.1. Curve

Because hatching is so multivalent, strictly defining a hatching curve is impossible. Its degree, curvature, and continuity are unstatable. It may intersect itself, be open or closed (Figure 15), or have gaps.

Sometimes the artist will have individual hatching marks trail off, such as by varying pen pressure during a single stroke. Along its length, a curve can widen or narrow, break up into gaps, or change in transparency (Figure 16). "Curve" is actually a misleading term. It is tempting to think of a hatching mark as a one-dimensional path, but even the possibility of this mark varying in width along its length makes a one-dimensional curve insufficient to represent it. The mark is not a path but a shape with area, and it may be fuzzy or discontiguous. The sides of this long shape could be ragged, especially in a drawing made with physical media or meant to look made with physical media. An artist might even refine an individual curve-shape by drawing other curve-shapes on top of it, creating something that reads as a curve to the eye but appears as a spiky mess to algorithms that take either raster- or vector-based input. Work in sketch beautification and simplification [BTS05, LRS18] is applicable here.

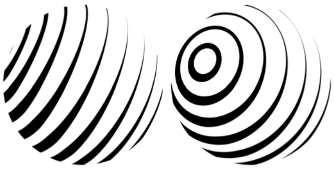


Figure 15: *Hatching curves can be closed. Otherwise, two of the curves on the right would be arbitrarily excluded.*

Figure 16: *Hatching marks can trail off by narrowing, breaking up, or fading.*

3.1.1. Hatching or stippling?

Figure 16 shows curves breaking up as a way to trail off. Past a point, breaking up curves produces a result that most would call stippling, not hatching. Figure 17 shows that it is hard to state when one becomes the other (unless one describes the two images as white hatching on top of black hatching). This is one simple way that identifying hatching becomes a necessarily indefinite task.

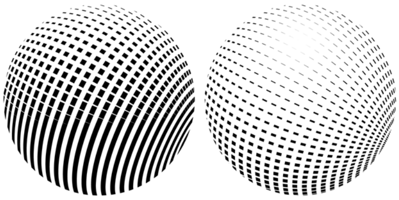


Figure 17: *Both scenes show a sphere lit by a single light. Changing the position of this light illustrates the fineness of the distinction between hatching and stippling.*

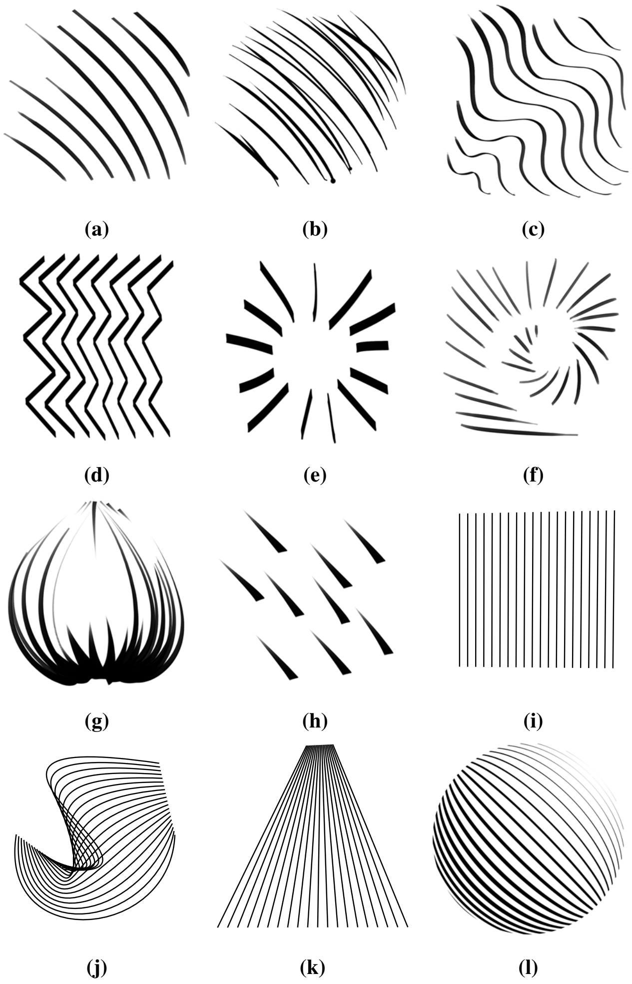


Figure 18: *Patches*

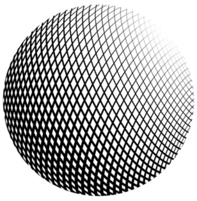


Figure 19: *Crosshatching*

3.2. Patch

A *patch* [GI13] is a single instance of hatching, a group of curves that run together and can be loosely ordered from first to last such that each curve is about parallel with the next (Figure 18). It may be useful to think of each curve in a patch as a parallel offset curve of the one before it, but this need not actually be the case. Curves in a patch can intersect, whether because of sloppiness in their placement (Figure 18b) or because the patch overlaps itself (Figure 18j).

When separate patches overlap, the result is crosshatching (Figure 19). This technique is very old and useful, but we refrain from investigating it in detail since our definition of hatching operates at a simpler level. In our definitional context, crosshatching is just a case of hatching involving more than one patch. Briefly though, inter-patch relationships are important and worth exploring further elsewhere, for one because artists often use overlapping patches to deal with complicated forms (Figure 20).

Figure 20: *The patch on the back of the hand is separate from the patch along the hand's heel, even though one, more highly curved patch could have done the job of both. Using multiple patches is often easier and faster. Detail from Alexander Browne's* Ars Pictoria *[Bro69], 1669, copper plate.*

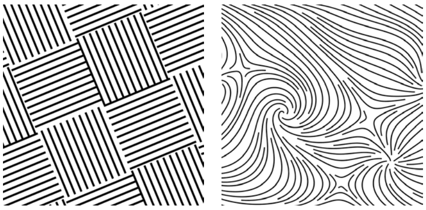


Figure 21: *Some patches are easily separable (a), others less so (b). Second image from Jobard and Lefer [JL97].*

3.2.1. Isolating patches

To even decide which curves in a drawing *might* constitute a patch is nontrivial. Our approach is to be generous. Figure 21 shows two images, each with multiple patches. The second is an illustration of a vector field using streamlines—not conceived of as hatching but in the spirit of the term as we see it. Not all its patches are easily separable. It might be worth clarifying that two patches should be allowed to share curves, thus making the streamlines image more comfortable to classify as hatching. Too much generosity produces absurdity, however. Simply grouping all the lines in a drawing that move in the same direction would seem to satisfy the definition of a patch just given, but such a disparate collection should not usually be considered a patch. This is where the last part of our definition of hatching becomes relevant: the marks in a patch should form a "flexible, simple pattern."

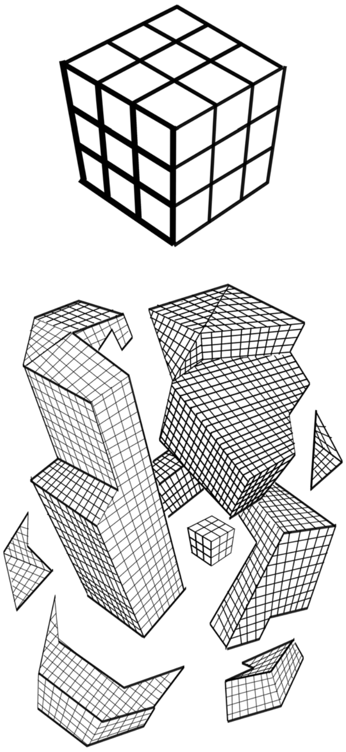


Figure 22: *In isolation, the cube's marks look too regular to be hatching, but this changes when the cube is in a context full of similar marks that are obviously hatching.*

3.3. Pattern flexibility

A patch should form a pattern that is not so rigid that any alteration of marks' directions, curvatures, density, or whatever else would change the semantics of the image.

Take the reticulated cube in Figure 22. Ostensibly, each of its sides contains two patches overlaid to form crosshatching. Yet the pattern is highly regular, and is easily read as the grooves in a Rubik's Cube. Changing the density of the hatching—the number of sub-cubes, in effect—would make the cube no longer a Rubik's Cube, thus changing the semantics of the scene. In other words, the pattern is inflexible. Context changes everything, however. If the same cube appears as a small part of a scene full of other reticulated forms, its own marks seem much less semantically important and more like a hatching pattern.

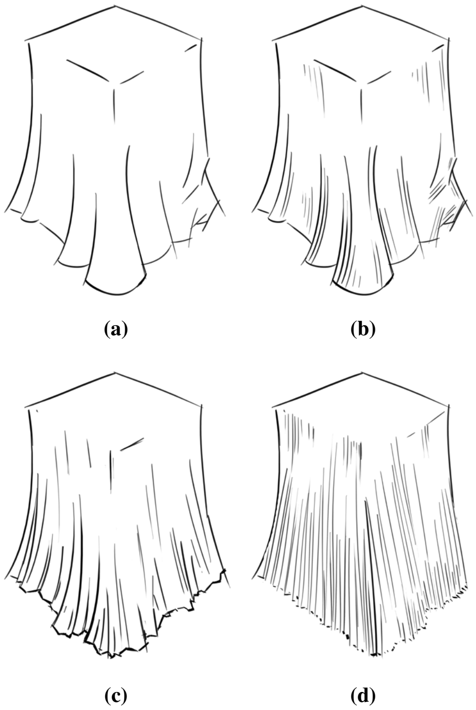


Figure 23: *If the folds are too large (a), it is hard to call their outlines hatching. However, additional marks (b) meant to reinforce the folds without defining new ones are hatching. When the folds are small enough (c,d), their outlines become hatching.*

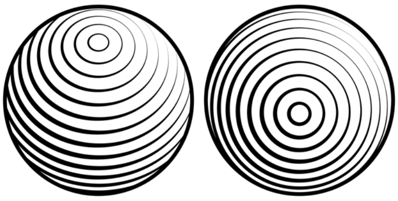


Figure 24: *Silhouette contours are customarily distinct from hatching (left), though a patch may be aligned such that it would be unreasonable to exclude a silhouette contour from it (right).*

Figure 23 is a testbed for the flexibility test: in each of the four illustrations, can we group all the approximately vertical marks in a single patch? In (a), there is a similar problem as with the cube example: the scale of the pattern is too coarse. Each contour on the inside of the tablecloth's silhouette is the side of a fold, and the folds are so large relative to the composition that they are almost forms in their own right rather than microforms on the surface of a macroform. There is another snag, one glossed over in the cube example. Two of the vertical marks currently up for consideration as members of a patch happen to be silhouette contours of the whole tablecloth, a form which is undeniably macroscopic. It is traditional to think of silhouette contours as a construct separate from hatching, though we do not feel strictly beholden to this distinction—see Figure 24.

Figure 23b introduces more marks next to each of the marks in (a), creating several different patches. For the same reason as in (a), it is still incorrect to group all the vertical marks as one patch, but the presence of hatching in the scene is now indisputable.

There is a more drastic change to get to (c). Now the scene geometry is different from that of (a)—perhaps the cloth is a different material—whereas the additional marks in (b) only clarify the forms in (a) rather than alter them. The folds in (c) are small enough

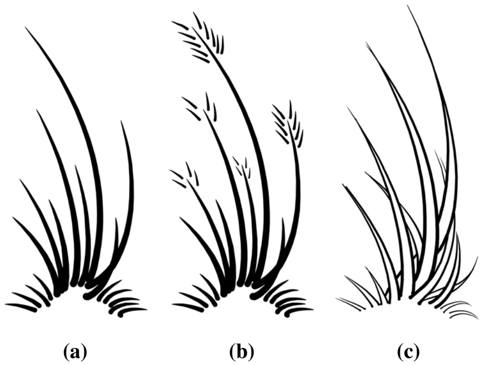


Figure 25: *When the curves of a single patch (a) start to relate to each other in particular ways (b,c), the patch is no longer definitely a patch.*

to enter the microform domain, making it easier to group the image's vertical marks into one or two patches.

In (d), the folds are even smaller. Yet the change is less significant. Moving from (a) to (c) means going from macro- to microforms. The transition from (c) to (d) does not involve such a phase change.

The flexibility test distills to this: First, a patch's curves must share a purpose (which means that to identify a patch is to apply the common fate principle from gestalt psychology). Second, the fewer the curves, the less confidently they can be regarded as a patch, especially when the specific number of curves appears to be important. Conversely, if it is possible to vary the number of curves in a group without affecting the purpose they serve or altering the image's semantics, then that group is more likely to be a patch.

3.4. Pattern simplicity

The previous test requires that curves in a patch work together in a flexible pattern. However, curves can cooperate too much, creating inter-curve patterns that make it harder to classify the group as a patch. To clarify, there is no problem with patterns contained inside each curve individually. There is no issue with a patch made of sine curves offset from each other, for instance, or with the patches of Figure 18c and Figure 18d. The problem occurs when one curve in a patch seems to have a special relationship with some other curve in the same patch, a relationship it does not have with others in the group.

Figure 25 shows three grass tussocks. It is easy to classify the first as a single patch. The second is trickier. The longer curves represent blades of grass while the smaller ones represent seeds. All the curves move together, but each seed curve has a relationship with a particular grass curve. It would be cleaner to call the grass blades a patch and each blade's group of seeds another two patches.

In the third tussock, inter-curve relationships are more numerous and particular. The curves outline individual blades of grass and honor occlusion relationships between the blades. There are special relationships wherever two curves outline a single blade or where one curve stops at its intersection with another in order to create occlusion. These relationships make it hard to call the whole tussock a patch, yet we cannot use the previous trick of partitioning the group into multiple patches either. Two curves outlining a single blade of grass are not a pattern—what would it mean to increase this pattern's density?

The simplicity test is highly subject to preconceptions about how complex hatching can be before it becomes something else. Both of the pattern tests should be taken with a grain of salt. They work best as a defense against totally spurious groupings of marks.

4. Expressive rendering

4.1. Creating hatching

There are three broad application areas in expressive rendering that involve generating hatching. One is rendering 3D models [WS94]. Then there are image-to-image applications where an input picture becomes a hatched illustration [SABS94, SWHS97, Ost99]. The third and least developed area contains applications like Vignette [KIZD12] and Strokes Maker, where the user directs hatching through curves drawn on-screen. These categories are fluid, of course: Kalnins et al.'s [KMM*02] application is technically a 3D renderer, but the user can draw hatching marks directly on the screen. See Lawonn et al. [LVPI18] for a survey of techniques for generating hatching.

The issues faced within these three areas are similar, with the exception of some concerns specific to 3D rendering.

Intuitively, software should outperform a human at hatching. For humans, creating patterns of curves requires carefully avoiding mistakes while exercising fine motor control over sustained periods. At the very least, we can expect software to draw a given patch far more quickly than any human. Yet with software there is a danger of creating hatching that looks too correct, sterile, or cheap. It is possible for a computer to generate a hatched version of any image in a facile way: a single patch covers the image, its curves' widths varying according to the image's local values (Figure 26a). The technique has no longevity; only the first few images generated with it will look impressive. A human would have difficulty making the same marks by hand, but this observation does not make the computer's output much more appealing. In this particular application, it is better to imitate a human artist, breaking the input image into areas of light, shadow, and form and choosing where to place patches according to this breakdown. Digital Facial Engraving [Ost99], in which an artist turns a photograph into a hatched illustration, lets the artist decide where patches should go and how their curves should bend (Figure 26b). To apply the same spirit in a different application, a 3D hatching renderer can imitate the old artistic practice of simplifying a scene's values into a small number of groups like highlights, midtones, and shadows, and then hatch consistently within these groups [HZ00].

Another danger in procedural hatching is moiré patterns. Figure 27 shows how crosshatching can cause these when the marks from different patches are too close to parallel. One solution is to imitate the irregularity of the human hand in placing and drawing individual marks, preventing unintended patterns from arising [VSE*06]. See AlMeraj et al. [AWI*09], who present a way to draw a curve between endpoints with anthropomorphic irregularity. An alternative approach is to base generated hatching on human examples [JEGPO02, KNBH12, GI13].

Imitating human output is a worthy goal in hatching applications generally. Understanding the human *process* of hatching is perhaps even more crucial for interactive applications where mouse or stylus input drives hatching. Recall the vagueness of the hatching curve definition in Section 3.1. This vagueness is necessary to encompass all marks considered hatching. A specific interactive application can use a much stricter definition. In the right context, one might assume that hatching marks come only from simple flicks of the wrist (a natural way to work, especially for those with figurative training). Under this assumption, we may define a hatching curve as an open, degree-2 Bézier curve (Figure 28). This point applies not just to the definition of curves, but to the whole definition of hatching. All its parts can become more technical and exclusive according to the needs of a particular research effort.

In 3D rendering, hatching marks often start out as object-space entities to be projected to image space. In Real-Time Hatching [PHWF01], a hatching mark originates as a shape in a texture that is mapped to an object's surface. In the sphere renderings of our paper, hatching marks are just procedurally generated stripes in *uv* space, which is a simplified version of how Knöppel et al. [KCPS15] produce hatching.

Originating hatching marks in object space like this is convenient because the object-world-camera transformation chain easily handles the complexity of bending them to convey form as in Figure 6b. However, hatching is ultimately a 2D design tool, and only when treated thus can it show all its capability, including its compositional applications. The Zorn etching of Figure 29 has multiple "lost edges," form silhouettes deliberately left out for compositional purposes like simplification and emphasis. The most (or rather, least) prominent of these is the edge between Zorn's jacket and his wife. This technique is fundamentally two-dimensional, and hatching contributes to it. A hatching renderer that deals with hatching marks as object-space curves cannot approach such techniques easily.

3D rendering has access to more of hatching's potential when it treats hatching as an image-space concern, informed by the 3D pipeline but operating outside it. Work is already moving in this direction. In the renderer of Lengyel et al. [LUS14], the placement of hatching marks occurs completely in screen space. The renderer of Orbay et al. [OK14] goes much further. It calculates screen-space shadow shapes to be flooded with hatching and fills them with marks whose paths are based on the shape they need to fill, not any 3D information. It even breaks up complex shadow shapes into separate patches the way a human might.

4.2. Analyzing hatching

The obvious motive for analyzing existing hatching is to improve procedural hatching, but there are other reasons. Most involve deriving a scene or object from a hatched illustration. The ability to generate a 3D form from hatching could be useful in product prototyping, for example. BlendFields [IBB15] and CrossShade [SBSS12] almost get to the point of producing form from hatching in the product design context. They process drawn silhouette contours and cross-contours to produce geometry. Whether the cross-contours can be called hatching is debatable, but certainly the process by which cross-contours guide geometry generation could apply to more conventional hatching.

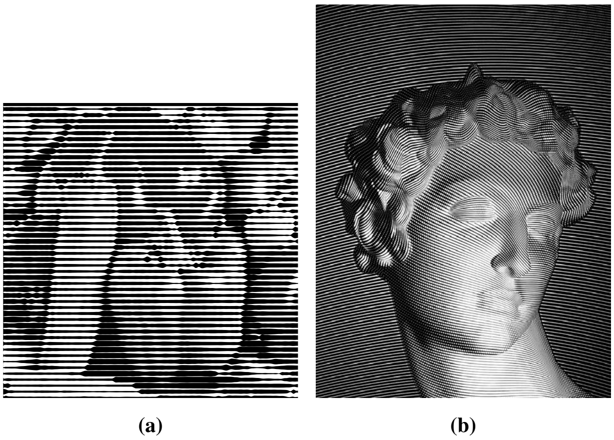


Figure 26: *The simplest way to turn any image into hatching is via a single patch with straight curves (a). It is better to break up the image into different patches whose curves have purposeful directions (b). The original for the first image comes from www.imageprocessingplace.com. The second is output from Digital Facial Engraving [Ost99].*

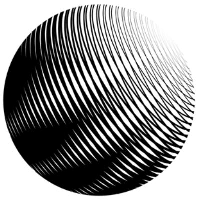


Figure 27: *Computer-generated hatching marks are susceptible to moiré patterns.*

Product prototypes are not the only forms to be inferred from hatching. Most hatched drawings contain inferable 3D information, and this information can be much more interesting than mere smooth surface geometry. A scene generator could read texture from hatching, perhaps in a scale-agnostic or scale-flexible manner, according to the loose way in which hatching communicates texture. Even if an algorithm that produces a 3D scene from a 2D

Figure 28: *Hatching marks made with flicks of the wrist can be treated as quadratic Bézier curves.*

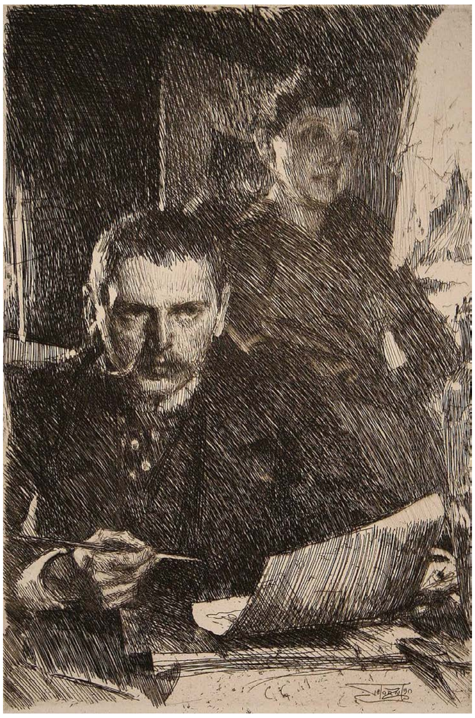


Figure 29: Zorn and His Wife*, Anders Zorn, 1890, etching*

illustration has no particular use for hatching, it will benefit from being able to properly detect and filter hatching marks.

Other, more unusual applications might lie in the future. Consider hatching's special utility to describe forms in medical illustrations [MSVF09]. One might use procedural hatching to create a medical illustration from a ground-truth 3D model, use a form-from-hatching algorithm to come up with a new 3D model, and measure the differences between the original and new model to study the reliability of either or both of the algorithms involved.

We can imagine algorithms at a more rarefied level that digest a hatched illustration to produce a "2D scene": a high-level breakdown of the image's pieces and an analysis of how the pieces work together, perhaps even an indication of why beauty emerges or does not emerge from the design. Such esoteric investigations would at least be useful for artists who hatch.

Whatever the application, software that detects or interprets hatching must be robust enough to deal with how messy hatching can be. Figure 30 contains hatching which could be particularly challenging for an algorithm to parse. Look closely and notice that some of its patches are actually single strokes, the result of hatching quickly without lifting the pen. Thus, a complicated back-and-forth curve might be best interpreted as a patch of simple curves glued together (Figure 31), or it might be a member of a patch along with other zigzag curves. Context must inform the decision.

When inferring 3D geometry from patches, there is a risk of perceiving depictive intent from patch-making decisions that are stylistic, meant to accommodate the artist's hand, or just mistakes. We have already seen Figure 20, in which the artist uses multiple patches where a single, more complicated one would suffice. An algorithm meant to infer form from hatching might erroneously perceive extra surface detail from patch multiplicity when it should instead account for incidental reasons why two patches might appear instead of one.

Figure 30: *Detail from* Der Faun*, Heinrich Kley, 1912, pen and ink*

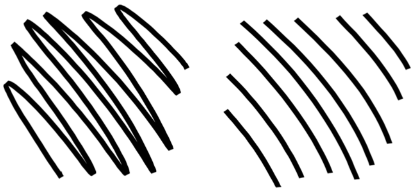


Figure 31: *We might interpret the curve at left as an entire patch, nearly equivalent to the patch at the right except made without lifting the pen between individual strokes.*

The challenge of analyzing hatching does not arise solely from the artist's haste or her other limitations. Some quite deliberate hatching decisions can pose difficulty, like in Figure 29, where Zorn produces lost edges using hatching marks that obscure shapes by moving across their boundaries. A sophisticated algorithm is necessary for properly reading the hatching of such an image without being confounded by its numerous individually misleading marks.

5. Conclusion

Hatching is art, and art resists taxonomy. Our definition of hatching cannot be authoritative. It is permissive enough to include marks that some artists would consider too complicated, like the streamlines in Figure 21; or too symbolic, like the black-on-white stars of Figure 11. Others might call our definition too restrictive. Rigidly labeling a subset of drawing marks is antithetical to art, which to some extent is about breaking rules. Yet expressive rendering often requires temporary guidelines and simplifications, and the ones given here are a useful starting point for work on hatching.

Acknowledgments

Thanks to Daniel Vogel for prompting this exploration, and to Linda Carson for an artist's perspective.

References

[AWI*09] AlMeraj Z., Wyvill B., Isenberg T., Gooch A. A., Guy R.: Automatically mimicking unique hand-drawn pencil lines. *Computers & Graphics 33*, 4 (2009), 496–508. 8

[Bro69] Browne A.: *Ars Pictoria*. Redmayne, Tooker, Battersby, 1669. 6

[BTS05] Barla P., Thollot J., Sillion F. X.: Geometric clustering for line drawing simplification. In *ACM SIGGRAPH Sketches* (New York, NY, USA, 2005), SIGGRAPH, ACM, pp. 183–192. `doi:10.1145/1187112.1187227`. 5

[Dod18] Dodgson N. A.: Abstract depiction of human and animal figures: Examples from two centuries of art and craft. In *Proc. Expressive '18* (New York, NY, USA, 2018), Expressive, ACM, pp. 8:1–8:8. `doi:10.1145/3229147.3229152`. 3

[Ens03] Enstice W.: *Drawing: Space, Form, and Expression*. Pearson College Division, 2003. 2, 3

[GDS04] Grabli S., Durand F., Sillion F. X.: Density measure for line-drawing simplification. In *Proc. Pacific Conference on Computer Graphics and Applications* (2004), IEEE, pp. 309–318. 3

[GI13] Gerl M., Isenberg T.: Interactive example-based hatching. *Computers & Graphics 37*, 1-2 (2013), 65–80. 6, 9

[GIHL00] Girshick A., Interrante V., Haker S., Lemoine T.: Line direction matters: An argument for the use of principal directions in 3D line drawings. In *Proc. Non-photorealistic Animation and Rendering* (New York, NY, USA, 2000), NPAR, ACM, pp. 43–52. `doi:10.1145/340916.340922`. 3

[Gur10] Gurney J.: *Color and Light: A Guide for the Realist Painter*, vol. 2. Andrews McMeel Publishing, 2010. 2

[Hux52] Huxley A.: The doors of perception. *Mental 98* (1952), 2–24. 5

[HZ00] Hertzmann A., Zorin D.: Illustrating smooth surfaces. In *Proc. Computer Graphics and Interactive Techniques* (New York, NY, USA, 2000), SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., pp. 517–526. `doi:10.1145/344779.345074`. 8

[IBB15] Iarussi E., Bommes D., Bousseau A.: Bendfields: Regularized curvature fields from rough concept sketches. *ACM Trans. Graph. 34*, 3 (May 2015), 24:1–24:16. `doi:10.1145/2710026`. 9

[JEGPO02] Jodoin P.-M., Epstein E., Granger-Piché M., Ostromoukhov V.: Hatching by example: A statistical approach. In *Proc. Non-photorealistic Animation and Rendering* (New York, NY, USA, 2002), NPAR, ACM, pp. 29–36. `doi:10.1145/508530.508536`. 9

[JL97] Jobard B., Lefer W.: Creating evenly-spaced streamlines of arbitrary density. In *Visualization in Scientific Computing* (Vienna, 1997), Lefer W., Grave M., (Eds.), Springer Vienna, pp. 43–55. 6

[KCPS15] Knöppel F., Crane K., Pinkall U., Schröder P.: Stripe patterns on surfaces. *ACM Trans. Graph. 34*, 4 (July 2015), 39:1–39:11. `doi:10.1145/2767000`. 9

[KIZD12] Kazi R. H., Igarashi T., Zhao S., Davis R.: Vignette: Interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *Proc. SIGCHI* (New York, NY, USA, 2012), CHI, ACM, pp. 1727–1736. `doi:10.1145/2207676.2208302`. 8

[KMM*02] Kalnins R. D., Markosian L., Meier B. J., Kowalski M. A., Lee J. C., Davidson P. L., Webb M., Hughes J. F., Finkelstein A.: WYSIWYG NPR: Drawing strokes directly on 3D models. *ACM Trans. Graph. 21*, 3 (July 2002), 755–762. `doi:10.1145/566654.566648`. 8

[KNBH12] Kalogerakis E., Nowrouzezahrai D., Breslav S., Hertzmann A.: Learning hatching for pen-and-ink illustration of surfaces. *ACM Trans. Graph. 31*, 1 (Feb. 2012), 1:1–1:17. `doi:10.1145/2077341.2077342`. 9

[LA08] Lau D. L., Arce G. R.: *Modern Digital Halftoning*. CRC Press, 2008. 2

[Loo12] Loomis A.: *Successful Drawing*. Titan Books, 2012. URL: `https://books.google.ca/books?id=LWp4AAAACAAJ`. 2

[LRS18] Liu C., Rosales E., Sheffer A.: StrokeAggregator: Consolidating raw sketches into artist-intended curve drawings. *ACM Trans. Graph. 37*, 4 (July 2018), 97:1–97:15. `doi:10.1145/3197517.3201314`. 5

[LUS14] Lengyel Z., Umenhoffer T., Szécsi L.: Realtime, coherent screen space hatching. In *Proc. Hungarian Computer Graphics and Geometry Conference* (2014), pp. 131–137. 9

[LVPI18] Lawonn K., Viola I., Preim B., Isenberg T.: A survey of surface-based illustrative rendering for visualization. *Computer Graphics Forum 37*, 6 (2018), 205–234. `doi:10.1111/cgf.13322`. 8

[MARI17] Martín D., Arroyo G., Rodríguez A., Isenberg T.: A survey of digital stippling. *Computers & Graphics 67* (2017), 24–44. `doi:https://doi.org/10.1016/j.cag.2017.05.001`. 2

[MSVF09] Medeiros J., Sousa M., Velho L., Freitas C.: Perspective contouring in illustrative visualization. In *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing* (Oct 2009), pp. 48–55. `doi:10.1109/SIBGRAPI.2009.49`. 3, 10

[OK14] Orbay G., Kara L. B.: Pencil-like sketch rendering of 3D scenes using trajectory planning and dynamic tracking. *Journal of Visual Languages & Computing 25*, 4 (2014), 481–493. 9

[Ost99] Ostromoukhov V.: Digital facial engraving. In *Proc. Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., pp. 417–424. `doi:10.1145/311535.311604`. 8, 9

[PHWF01] Praun E., Hoppe H., Webb M., Finkelstein A.: Real-time hatching. In *Proc. Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH, ACM, pp. 579–584. `doi:10.1145/383259.383328`. 9

[SABS94] Salisbury M. P., Anderson S. E., Barzel R., Salesin D. H.: Interactive pen-and-ink illustration. In *Proc. Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH, ACM, pp. 101–108. `doi:10.1145/192161.192185`. 8

[SBSS12] Shao C., Bousseau A., Sheffer A., Singh K.: Crossshade: Shading concept sketches using cross-section curves. *ACM Trans. Graph. 31*, 4 (July 2012), 45:1–45:11. `doi:10.1145/2185520.2185541`. 9

[SWHS97] Salisbury M. P., Wong M. T., Hughes J. F., Salesin D. H.: Orientable textures for image-based pen-and-ink illustration. In *Proc. Computer Graphics and Interactive Techniques* (New York, NY, USA, 1997), SIGGRAPH, ACM Press/Addison-Wesley Publishing Co., pp. 401–406. `doi:10.1145/258734.258890`. 8

[VSE*06] Viola I., Sousa M. C., Ebert D. S., Andrews B., Gooch B., Tietjen C.: Illustrative visualization for medicine and science. In *Eurographics (Tutorials)* (2006), Citeseer, pp. 1061–1200. 3, 8

[WS94] Winkenbach G., Salesin D. H.: Computer-generated pen-and-ink illustration. In *Proc. Computer Graphics and Interactive Techniques* (New York, NY, USA, 1994), SIGGRAPH, ACM, pp. 91–100. `doi:10.1145/192161.192184`. 3, 8