Expressive Rendering: A Review of Nonphotorealistic Techniques

For some time the main thrust in computer graphics research has been toward production of images indistinguishable from photographs of actual objects and scenes. Recent developments in virtual reality have added impetus to achieving photorealism. The extent to which researchers have succeeded in this aim can now be judged by viewing some feature films and TV commercials, where it is often impossible to decide whether you are seeing pictures of a simulation or the real thing.

However, at the Centre for Electronic Arts, Middlesex University, and the Martin Centre CAD Laboratory, University of Cambridge, interest has focused not only on making photorealistic images but also on rendering objects and scenes to resemble more how artists and designers—rather than snapshot photographers—might want to see them. The resulting images may encourage viewers to make the same imaginative, perceptual contributions they make to interpretive art.

The default tendency of nonphotorealistic rendering (NPR) systems is to generate imagery that superficially looks like that made by artists—for example, pen and wash, water-color, or charcoal. Some NPR systems emulate an artist’s handwork, but most of these are prototypes. The systems that have found their way into commercial use are effectively image processors or paint systems.

We feel, however, that the most interesting developments will emerge when NPR systems are, in a functional sense, much more like complete 3D renderers—capable of interpreting all the data available in a 3D model in order to generate images that possess some internal meaning and structure. To this end, one of the authors, Simon Schofield, has developed a prototype system that gives users a wide variety of automatic rendering styles—from the almost photorealistic, through the styles resembling artists’ sketches or paintings, to abstract renderings of entirely new forms (see Figure 1, next page).

To us, the significant problems associated with NPR are primarily aesthetic and only secondarily technical. The culturally oriented theoretical problems surrounding the field are many and, at their most difficult, broach central issues of perception and representation. We therefore begin by discussing the limitations of photorealism as a representational style.

However, the technical problems of implementing an NPR system are not trivial. As far as we know, the literature defines no general framework for an NPR system, although it does include an array of more focused solutions. We survey more than 15 of these solutions in the second part of the article. On the basis of this survey, we then define a framework for NPR systems that distinguishes them from systems focused on photorealism. We also identify the characteristics of NPR systems that go beyond the tendency to simulate traditional media, functioning instead as 3D renderers. In this context, we briefly describe Schofield’s system, called Piranesi, which operates this way.

Photorealism: One representational form

Photorealism enjoys a privileged position in the universe of images. This is strange in that historically it has only rarely been thought necessary for artists to represent the external world in an “objective” way. Gombrich has gone to some lengths to counteract unquestioning regard for the sanctity of the photograph. He challenges the presumption that photography is a correct or “direct” form of representation, more intimately bonded to the reality it represents than other image forms. He shows how heavily abstracted a photograph actually is through exposing the many artifices employed: the micro-instant frozen forever, the limited angle of view, and the arbitrariness of photographic processing. However, since the Renaissance, Western art has tended toward repre...
senting selective portions of the world with accuracy to its material/physical appearance. This tendency was affected by but not totally abandoned after the development of photography. The nature and attendant successes of such art have led many people to regard photorealism as “more correct” than other forms of representation, and hence more desirable.

If we look at a selection of computer-graphic photorealistic images produced over a period as brief as the past five years, we are struck by the vast improvements made in that time. This is to the credit of research in the field. Nevertheless, in our view, it is still extremely difficult to wrest visually pleasing images from most rendering systems. A blandness or sterility often shows up in these images. The generally accepted method for addressing this problem is to increase the intricacy of synthetic images until they approach the visual complexity of real scenes. The means to this end include antialiasing, multiple light sources, soft shadows, and texture-, bump-, environment-, and displacement-mapping. Other techniques such as simulated depth of field, lens flare, and motion blurring can also be applied to the image. Intriguingly, these effects are sometimes unwanted artifacts of conventional photography, yet they are deliberately introduced into computer graphic scenes to convince us of their “photographic” veracity.

When viewed from a broader cultural perspective, photorealism is only one possible pictorial style among many. The use of alternative and more radical forms of representation is particularly strong in art and design, where photorealism can sometimes be less than desirable. But it is true even in other fields. For example, consider the information displayed in, say, the Bayeux Tapestry with that in the mass of photographs of the First World War. Both selectively represent reality. But one undoubtedly serves historical interpretation differently from the other. The historian can only infer so much from the photographs—a mass of specifics without overall coherence. The tapestry reveals a story through its generality supported by some visual specifics.

In another context, consider the value of a photograph of a car engine in a maintenance manual compared with that of an exploded schematic line drawing. How much use is a photograph to mechanics when they already have the real thing in front of them? Photographs often perform poorly when clear and precise delineation, explanation, and understanding is required. It is not surprising then that other representational methods are used when planning and designing things. The traditional tools of pen and pencil, standard legends and notation, agreed conventions and certain methods of presentation, such as the projections used...
in working drawings, serve many purposes better than any amount of photographic documentation. Furthermore, photorealism is particularly poor at dealing with the general because it insists on the specificity of the referent. It implicitly insists that the image presents all the detail originally in the referent, subject to restrictions of focus and grain-size. Hence, within these limits, a photograph can be enlarged to reveal further detail. As a consequence, photorealistic imagery possesses an inherent suggestion of completeness. We are not used to looking at "partial photography," where the scene is not complete. Thus, we often fail to question the documentary nature of photographs.

**NPR techniques and systems**

Computer graphics research includes many interesting attempts to widen the range of rendering output possibilities beyond photorealism. Broadly, two main approaches exist, categorized as using either image space or perspective space effects. We describe some of these NPR techniques and systems within these categories.

**Image space effects**

The effects in this category are operable on almost any image.

**Mezei's texture elements.** As far back as 1974, Mezei et al. presented a technique for generating images that bear some relation to hand-drawn illustrations. Initially, the authors intended the technique to synthesize patterns seen in nature, but they suggested it could also be used to achieve high degrees of natural realism. This pioneering effort presaged most current nonphotorealistic systems in the way it formed an image by applying a mass of prefigured graphical subunits. Its underlying idea is that many things in nature are composed of visually distinct elements arranged in various distributions. These subunits, or "texture elements," can be synthesized and a range of effects achieved by resizing, rotating, and otherwise perturbing them before applying them to an image. Probabilistic algorithms control the distribution of subunits across a surface.

**Hairy brushes.** Strassman outlined a technique that stems from his interest in modeling fuzzy, complex systems and focuses very much on behavioral simulation of paintbrushes and ink. He used object-oriented techniques to create sets of rules and classes of objects that accurately model the way a wet brush works. His brush has a one-dimensional footprint of bristles swept perpendicular to and along a spline. This spline is defined by a number of nodes, each containing the position and brush pressure at that point. The pressure is linearly interpolated between nodes. Corners and turns in the stroke path are negotiated by rotating the bristles around the inner edge of the mark. Each stroke is laden with a specific amount of "ink," which is exhausted along the length of the stroke. Users refill the ink by "dipping," and this interruption lets them alter the quality of the marks from stroke to stroke. The medium is not restricted to just one shade. It supports a range of tones changing over the width of the footprint, thereby imitating the subtler nuances of Japanese calligraphic painting. Users can further enhance internal texturing within one mark by texture-mapping.

**Iterated Function Systems.** Barnsley's work on IFS fractals generated several convincing painterly renderings of images. The renderings are derived from digitized photographic images, although Lansdown demonstrated a design use of IFS that might have relevance to NPR by allowing controlled changes in the structure of objects in an image. Barnsley's well-known picture of a Bolivian girl wearing a hat is particularly impressive (reproduced in the referenced article and on the cover of *Fractals Everywhere*). This image is derived by IFS from a photograph but looks as if it were painted—perhaps by Gauguin. Features in the image, such as the eyes and the pattern around the hat, have become somewhat exaggerated, presumably as a result of the IFS self-similarity and smoothing operations. The overall effect is haunting but seems to caricature the head in a way that a cartoonist might. Barnsley's main concern, however, is in the high data-compression ratios offered by algorithmic encoding that uses fractal, or more specifically, IFS methodologies. The long-term aim is the real-time compression and decompression of photographic imagery for broadcasting. Hence, the visual distortions attractive to an NPR enthusiast might be viewed by others as unwanted artifacts. The compression technique relies on a mathematical concept called the "collage theorem," as well as the user's ability interactively to help the algorithm make good choices of affine transformations.

The collage theorem, which relies on the inherent self-similarity of all images, apparently guarantees that any IFS using the proper affine transformation will regenerate a picture that resembles the original. Although this technique would be very appropriate for obviously self-similar images (leaves, mountain ranges, and so on), Barnsley and his co-workers must get credit for recognizing the universality of the method as a way of approximating any original. Unfortunately, the technique works more effectively on some images than others, and the actual transformations used for the Bolivian girl's head heavily distort the end image even though much of the encryption for the image has, we assume, been defined by hand.

As Barnsley stated, "The closeness of this rendition to the desired image depends on the accuracy with which the user is able to solve interactively a certain geometric problem," although just what this involves is unclear. Barnsley claims that the interactive part of solving geometric problems can be automated, and this would be necessary for the compression technique to be generally useful.

**Haeberli's systems.** Paul Haeberli described several techniques for generating "painterly" renderings of digitized images and 3D geometries. First, frame-buffered images are again repainted using a range of marks, each created from a moderately simple graphic element that samples corresponding points in the image...
buffer and recolors as necessary. Using a range of different marks within the same image, this technique achieves "pointillist" painting styles. The marks are parameterized in respect of their location, color, size, orientation, and shape.

Haeberli's systems include a user interface for real-time hand interaction with the system. Users can guide and direct strokes over the blank "canvas," and change the size and direction of the strokes. They can also focus the activity of painting in several ways. In one implementation, pressing the mouse button causes randomly placed large marks to tighten up around the mouse position; as they do so, the marks get smaller. An alternative to this is to make the tightness of focus of the marks and their size inversely proportional to the speed of the mouse movement. Hence, rapid movements of the mouse cause large widespread changes, while slow movements cause highly localized changes.

There are also tools for automating the production of paintings. A brush-stroke display list can be generated and reused to generate a sequence of visually consistent images from a sequence of frame-grabbed images. The automatic reorientation of strokes over an image is implemented by establishing relationships between tone gradient and brush direction. The tonal gradient can be calculated from the original image or from a separate underlying luminance map.

Haeberli achieved the interactive painterly rendering of 3D geometries by replacing the simple digitized image of the system described above with a screen-sized port into a ray tracer. Object color and surface geometry are calculated in real time by firing rays into the scene at the mouse location rather than by sampling pixel color values. This method makes surface geometry and other material properties available for various uses. The geometry automatically reorients the direction and size of stroke depending on surface normal and distance from eye to surface. Marks can be made to accentuate object geometry by orienting them along lines of curvature.

Haeberli presented a separate technique for creating pointillist renderings through the use of Dirichlet domains (or Voronoi regions). An iterative searching technique is applied to these areas, adjusting them to fit and color themselves to a frame-buffered image. Initially, the randomly placed polygonal shapes color themselves to the average of all the pixels in the corresponding source-image patch. Over a sequence of iterations, the shapes relax into best-fit configurations, trying to represent the underlying color by reducing the standard deviation of the averaged color to a minimum. The configurations are constrained by the fact that the initial number of Dirichlet domains cannot change and that the total surface area of the patches must remain constant (that is, the patches must cover the whole surface).

"Wet and sticky" model. This approach uses a cellular automata approach to model a paint-like substance and apply it to a canvas substrate. 21 Cockshott et al. 21 extended the system to exploit the way a cellular automaton structure might be employed to generate a bump map. The results create a convincing 3D textured paint-like surface. Cockshott's doctoral dissertation presented a method for creating a paint-like substance that shares many characteristics of natural wet and sticky substances when applied to a canvas support. The substance bleeds, runs, drips, and mixes. It dries out and can change appearance as it does so. It responds to "gravity" as well as other environmental conditions such as temperature and the nature of the support.

The system uses paint redistribution processes performed by its painting engine. One of these processes averages quantities of paint with surrounding cells by spreading the load of stored paint particles as evenly as possible within a set time before the paint "dries." This causes the paint to bleed and mix. The movement of paint around the canvas is determined by sets of rules designed to simulate environmental conditions such as ambient temperature, which affects the drying speed of paint, and gravity, which defines the direction of dripping. While the system performs its task automatically and as quickly as possible, it is designed for users to apply the original paint via normal interactive techniques. Because of the heavy processing required for real-time dripping and bleeding, Cockshott advocates parallel computing techniques.

Wobbly pens. Van Bakergen and Obata 22 employed the simple and amusing technique of attaching loose, wobbly pens to a pen plotter. This produces surprising and effective results. A survey of what people think of these new "re-humanized" sketches showed that students who previously fought shy of including any computer graphics in their portfolios felt no inhibitions about including these wobbly plots.

Van Bakergen and Obata outlined several possible explanations for the attraction of the wobbly images. One is that they are unique: The deviations of the pen cannot be reproduced. Another is that they are simply more visually engaging: They seem to have more detail—or at least more to look at. The images benefit from a synergy between the absolute control of the machine and the randomness of the wobble, the result being a sketch that seems more precise than freehand productions. These "freehand plots" have apparently duped experienced draftsmen into believing them to be very competent hand drawings. As a result of these investigations, van Bakergen and Obata produced a PostScript program that provides more control over the effects of manipulating plot files.

Quantel Paint Box. Probably the first available explicitly painterly technique for any paint system was Quantel’s “chalk” effect on the paintbrush menu of its Paint Box. This effect was achieved by a simple lookup table deciding whether or not certain pixels ought to be omitted when painting a mark.

ImagePaint. ImagePaint (ImageWare Systems, Toronto, Canada) also predates most well-known painterly paint systems by several years. An interactive postprocessing package, ImagePaint applies a series of filters to an image. It subdivides the image into many overlapping subimages, filters the subimages, and finally adds them to the output image with varying degrees of opacity. The
original image is subdivided by defining various meshes and pulling parts of the image through them. These filter meshes can be regular or random and can be combined with blanket effects such as edge-finding.

By storing the meshes and filters as a set of parameters, users can apply them to a sequence of images. This facilitates some animation techniques, although for perceptual reasons the results are not as effective as those on still images.

**Render Strokes.** Pacific Data Image developed an in-house system called "Render Strokes" for postprocessing photorealistic images to create painterly images. The company has published no implementation details, although manual techniques for editing individual frames by using various filters and brushes would produce the same results. We assume that part of the post-processing happens through noninteractive techniques.

**Mitchell's method.** Mitchell demonstrated some image processing techniques that use a basic image-processing package to generate painterly images from digitized images. The techniques are a combination of one-to-one pixel-mapping routines and random noise functions applied automatically across the whole image. Posterization and edge-detection produce watercolor-like images. Random or wave functions can add noise to the image.

**Fractal Design Painter.** Fractal Design Painter (Fractal Design, Aptom, Calif.) is an interactive 2D painting system that specializes in "natural media" effects by emulating real-world media such as charcoal, ink, pencil, oil paint, and so on. It uses a range of painting tools, brushes, filters, and textures. Many of these techniques employ types of randomness to "naturalize" the medium. For instance, small cycles through prefigured lookup tables can generate the grittiness of charcoal.

Users apply the techniques through standard interactive painting methods, although they also have the option of selecting large areas for blanket effects. Mixing these naturalistic effects in certain ways can result in new image styles that bear little relationship to traditional techniques. There is also some facility for automatic painting. For example, a "cloning" process re-renders grabbed images from a stored script. Other interesting and novel facilities include the ability to push "wet" paint around on the surface with a "palette knife."

The emphasis on emulating natural media—that the end product should look like a scanned oil painting, for instance—shows up in the inclusion of surface embossing effects. These emulate the positioning of lights at raking angles to the surface, thereby highlighting the tooth or weave of the canvas and the impasto effects of oil paint. This is probably achieved by embossing through convolution filters, then multiplying the luminance of the original image with the embossed image. Other more elaborate filters distort the image as if placing it under rippled glass. These effects are achieved by using spatially coherent 2D noise generators to displace colors from their original positions. Fractal Painter embodies an interactive paint package solution to many of the problems posed by the challenge of simulating natural media.

**Other approaches.** Several other noteworthy packages display features within a more conventional paint-package framework. UltraPaint (Deneba Systems, Miami, Fla.) has an excellent blending algorithm for layers of over-painting. The results get very close to the look of ink or water color on paper. Photoshop (Adobe Systems, Mountain View, Calif.) has a set of plug-in filters, such as Aldus Gallery Effects by ImageWare R&D (Aldus Corp., Seattle, Wash.), that process selected regions of an image to create various pointillist styles.

**Perspective space effects**

These effects require more than a simple image to work properly.

**Sasada's landscape system.** Sasada tries to emulate graphical conventions seen in technical drawing and illustration. His work concentrates on the different types of cross-hatching seen in artists' illustrations of landscapes, skies, and seascapes. He wants to reduce the computational effort required to render landscapes when making animations of cities in their geographical context. Most of his algorithms start with a standard hidden-line rendering of a regular mesh that represents the mountains. Then either the rendering is stochastically perturbed or sharp changes in mesh direction are processed to keep only ridges, thus creating a natural look.

A convincing hand-drawn technique emerges from calculating "water-flow" lines instead of a simple mesh. Water-flow lines are meshes of lines whose axes are defined as orthogonal to a mountain's altitude contours, that is, directions in which water would flow down a hillside. Different line thicknesses emphasize light and shade.

For closer views of mountains and natural scenery, Sasada uses 2.5D representations of trees and foliage to cover the landscape. These are 2D vector representations of trees that are shifted, scaled, rotated to face the point of view, then drawn with varying line weights to represent forests. These techniques are not as convincing as the more distant mountain views because the repetition of more complex graphic elements is too apparent. Furthermore, animation can betray the fundamentally 2D nature of the representation. To solve the first problem, Sasada suggests using automatic stochastic tree pattern generation of the sort devised by Lindenmayer or Aono and Kunii.

**"Comprehensible" rendering techniques.** Saito and Takahashi presented several techniques for increasing the comprehensibility of 3D rendered shapes. Their premise is that most technical illustrations are clearer than photographic representations through their use of edges and hatching. Their system adds this extra layer of delineation on top of photorealistic images, first by rendering a scene into a "geometry buffer" that stores the 3-space coordinates as an image-resolution raster. This can then be sampled on a per-pixel basis and image-processed to create various interpretations of the original scene.
Sobel edge detection on the z-field of each pixel produces good line drawings by finding discontinuities. Profile edges (silhouettes) and internal edges (corners) can be found separately and rendered using different filters. Generally, a good drawn effect is achieved by weighting the density of the pixel to the level of discontinuity detected. Hence, silhouettes are shown as strong lines, whereas internal edges are finer—a technique often used in conventional sketching. The resulting image can then be used alone or reapplied to the photorealistic image to give it more clarity.

Hatching requires development of a height-contouring algorithm. This is a kind of solid-texturing method that striates the scene at regular pixel intervals in world space by using a thinning technique to constrain the line density in screen space.

**Strokes-Interpreted Animation Sequences.** Van Berkal designed the SIAS system to animate 3D models composed of painterly brush strokes; the strokes themselves exist in virtual 3-space. The system not only generates marks but also allows the resulting shapes to be animated in response to sound inputs (for such things as lip sync). In the video “Piece for 32 Nonexisting Voices,” stroke-composed animated heads articulate with realistic facial movements in synchronization with a synthetically produced voice. The system gives great consideration to accurate correspondence between the phoneme and the facial expression.

Following Strassman, van Berkal generates marks along a predefined spline, though in van Berkal’s case the spline is modeled in 3-space and projected into 2-space for mark-rendering. Once mapped onto the 2D spline, a polygon defining the shape of the whole mark is filled with parallel bands of color. These bands can change over their own length as well as over the length of the stroke. Some problems occur during filling when the mark polygon is oriented in certain directions. This is because the filling technique starts to split up to form highly visible bands.

**KATY.** This product is an automated scan-line renderer (5D Inc., London). It can produce technical illustrations using a range of photorealistic, graphic, and painterly techniques. Many KATY techniques are implementations of the Saito and Takahashi algorithms for increasing comprehensibility, although KATY employs a simpler technique for generating some flatter styles of hatching. Specifically, it calculates the luminance of a point in the output image using cosine shading and uses a mathematically defined, screen-consistent hatch to render that point. The hatch is defined by intersecting sine waves thresholded to two values—black and white. Altering the threshold alters the hatch density. KATY also supports automatic postprocessing of images along the lines of Haeberli and Fractal Design Painter.

**Stroke textures.** Winkenbank and Salesin have designed a system to produce drawings that resemble artists’ pen-and-ink sketches. The user employs a conventional 2D drawing system to create vector-based sketch textures for elements such as brickwork or grass. These elements are then stored as a display list and applied, very much like a conventional texture map, to a 3D scene—in full perspective. The process is enhanced by allowing the user to define the priority of certain marks over others. Thus the texture, when rendered far away or when foreshortened, will appear with less detail. This avoids unnecessary crowding of texture, which is a common and unwanted occurrence in conventional hidden-line drawing.

The system can also vary line densities in response to tonal variations in the 3D scene. “Detail tags” can be set in the 3D scene to define points of interest for extra rendering attention. The result is a convincing sketch rendering of textured surfaces in perspective with varying levels of detail that can help guide the viewer’s attention to areas of specific importance or interest.

**NPR framework for further development.**

Our survey of NPR systems indicates the breadth and diversity of research and solutions addressing the relevant issues. Despite this diversity, we can define the morphology of NPR by a relatively few distinct techniques:

- Images are constructed from indivisible pictorial subunits that are larger than the pixel, such as “brush marks” or polygons.
- Randomness or arbitrariness is included as a factor within the rendering process to create distortions and noise of various sorts.
- Techniques such as edge detection and hatching are included to create per-pixel scene artifacts that are not intended to be read as physical parts of the model but as part of the image.
- Most important is the way some NPR systems work in screen space and others in object space.

Artifacts used to compose the image do not easily map back into the 3D world. Rather, they have a stronger relationship with the 2D surface. The mark, image-processing, and noise features of Fractal Design Painter and Haeberli’s systems are applied to the image as 2D effects, while the edge drawing and hatching of Sasada, Saito and Takahashi, and the KATY renderer—though mapping more directly to the 3D model—are constrained by screen-space algorithms relating to width and density of the marks in 2D. An aspect common to all these systems then is the continual reexamination, and further complication, of the relationship between the 3D world space of the model and the 2D screen space of the image.

Most notable among the tendencies revealed among the systems surveyed is the assumption that NPR is synonymous with natural media simulation. This stance follows the general tendency of new technologies in their formative stages to copy older ones. The apparent intention to deceive by the pretense that these computer-generated images are pictures of real paintings and drawings, however, causes problems for many artists and designers. More importantly, new styles of presentation might emerge if we abandoned the reactionary and overly limiting approach of imitation.

To address these objections, we suggest sidelining the pursuit of natural media simulation. System designers...
can do this by eliminating certain development strategies, specifically those for techniques designed to convince viewers that system-generated images really were, at some time, natural media images. This means that faking the impasto surface of oil painting, or simulating the action of gravity on paint, should not be a development concern. Obviously, it does not mean discontinuing the development of representational devices that occur in natural media but transcend it (for example, color).

If both photorealistic and natural media imitation methods are ignored, the desirable NPR system must be able to:

- make marks in the scene using a choice of graphic elements,
- control the placement and style of these marks in response to the scene data,
- fire-off new sketching algorithms automatically when certain conditions are met,
- split the scene into discrete regions that allow application of a certain sketching method to a certain part of the scene,
- build a sketching algorithm that places the sort of mark desired in the correct place, and
- provide methods for automatic control and user interaction.

The Piranesi system incorporates these characteristics to a large extent. We describe it briefly here and in detail elsewhere.²⁰

**Piranesi system**

The Piranesi system has abandoned natural media simulation per se in favor of simple pictorial mark-making techniques. These techniques may bear a superficial resemblance to those used in traditional media, but simulating this resemblance is not their primary intention. For comparison, consider 2D and 3D hatching: While first used by engravers, hatching is a practical way of simultaneously half-toning and delineating form in any medium. Some other techniques employed by Piranesi, such as applying cutout shapes to the scene, more resemble computer-painting techniques than those of traditional media.

Initially called Expressive Marks,²¹ Piranesi was originally designed to automatically render painterly images from conventional 3D CAD models. These rendering techniques arose from the modeling of parameterized marks possessing a complex internal structure that, if desired, could be set to look like certain natural media—charcoal, pencil, felt-pen, crayon, ink, and so on (see Figure 2). However, this simulation was soon abandoned in favor of much more arbitrary methods of image construction. Because these methods are not constrained to imitate natural media, they open a whole new range of techniques for making marks.

Although it is a renderer, the Piranesi system looks like a paint system to the user. It has region application tools, painting tools, area selection tools, and a zoomable/scrollable painting window. These are used in conjunction with a table of rendering options and sliders for dynamically setting the properties of the chosen render effect. A rendering style is loaded onto a brush or flood-fill tool and applied to the image. While many of these operations require user interaction, high degrees of automated image production are possible. Some of these are suitable for generating sequences of images for animation.

Piranesi uses a standard rendering pipeline to generate source files. The RGB rendering, z-map, and a “material” channel are captured and stored as an Extended Pixel (EPix) image. The z-map, along with the original viewing transform, can be used to recalculate world-space geometry on a per-pixel basis with the necessary speed to allow real-time interactive operations. The process uses a copy of the RGB field of the EPix image or a Sobel edge-detection on the z-field as starting points for the rendered output. Users can load preset rendering styles from file, or build and tweak styles dynamically. Users select areas for rendering by specifying material type, surface normal, or planarity with a picking tool.

The rendering is applied to the chosen area using an automatic, semi-automatic, or interactive application tool. An automatic point visitation process based on a digital-dissolve effect applies a given rendering technique evenly over the selected region. Direct painting of rendering effects is also possible using a standard paintbrush tool. Users can erase effects by restoring from file. If compositing techniques modify the source image, users can save the updated EPix source image also. The best results are produced by overlayering techniques to
In doing so, Piranesi does not disperse with the inherent advantages of 3D rendering systems: correct perspective, good tonal approximation, and adequate detail (see Figure 3). In use, Piranesi "looks" much like a conventional paint system, but "feels" very different. The feel is of an interactive renderer that does "three-dimensional painting" (see Figure 4).

**Conclusions**

NPR strategies for image-making have developed against a background of graphics research dominated by the pursuit of photorealism. We suggest that photorealism is wholly concerned with the credible projection of 3D artifacts into 2D artifacts, so that everything seen in the image is supposed to "exist" in 3D also. By comparison, NPR is much more concerned with the image plane in isolation from the 3D scene it represents. NPR techniques employ a tendency to impose 2D artifacts in the image plane that clearly do not exist in world space (for example, marks, hatches, and drawn edges).

As our review shows, several excellent NPR techniques and systems exist. However, many problems must be solved before NPR can be considered a mature representational tool. We look forward to NPR systems that can claim to make "significant" images—that is, images embodying new levels of meaning rather than simply presenting an optical distortion of a previously photorealistic image. To do this, the systems must have more information about the scene they wish to depict,
primarily so they can distinguish one part of the scene from another. Piranesi can make this differentiation to some extent.

Piranesi has also helped explore the dynamics of human-computer interaction and user acceptance of images. For example, its output has aroused considerable interest in the architectural community, where the use of computer-aided design is widespread, while the reaction to photorealistic rendering is somewhat negative. That Piranesi will accept the sort of models created by architectural CAD systems is a bonus.

Finally, Piranesi has indicated that a more rewarding approach to computer graphics in art and design might lie somewhere between automatic rendering and interactive painting. However, our discussions with designers and architects have also revealed that they would not exclude any of the photorealistic technologies they currently enjoy from an NPR system. For this reason, NPR might be considered a superset of photorealism and not just an alternative to it.

References

John Lansdown is professor of computer-aided art and design at the Centre for Electronic Arts, Middlesex University, London. He was a founder of the Computer Arts Society and its honorary secretary from 1968 to its closing in 1991. He is a fellow of the Royal Institute of British Architects, a fellow of the British Computer Society, and a fellow of the Royal Society of Arts.

Simon Schofield is a research assistant at the Martin Centre CA disillusioned in the Department of Architecture, University of Cambridge, UK. (The Martin Centre CA disillusioned is funded by Informatix, Tokyo.) Schofield's current interests are nonphotorealistic rendering, interactive rendering of 3D scenes, 3D painting, and architectural representation. He studied fine art at Brighton University (1985), computing in art and design at Middlesex University (1989), and received a PhD from Middlesex University in 1994.

Readers may contact Lansdown at the Centre for Electronic Arts, Faculty of Art, Design and Performing Arts, Middlesex University, Cat Hill, Barnet, Herts, EN4 8HT, UK, e-mail: john17@mdx.ac.uk, and Simon Schofield via e-mail at ss152@cus.cam.ac.uk.