The Piranesi system for interactive rendering

Paul Richens

The Martin Centre for Architectural and Urban Studies,University of Cambridge Department of Architecture, UK

Key words: computer graphics, non-photorealism, painting, interaction, rendering

Abstract: Photorealistic rendering requires special skills to achieve, but the result is not necessarily ideal for architectural communication. The Piranesi software provides an alternative way of producing a much wider range of images from the same geometrical model. Piranesi appears to be a 2d paint program for editing raster images, but its input includes a z-buffer. This allows painting and and pasting to reflect the perspective of the image, and many other depth-related effects. The result is a new, and enjoyable, way of producing architectural images.

1. INTRODUCTION

1.1 Photorealism

Computer rendering has come a long way in the last twenty years. But is it going in the right direction? Is the glossy photo-realistic image the only goal worth pursuing? And does the process of making it contribute enough to the design, or the ongoing dialogue with the client?

There certainly are alternative modes of image-making. Frank Lloyd Wright, according to legend, could conceive a whole building in his head, and set it down rapidly, in plan and section. He would leave these drawings overnight with his assistant, who would set up a perspective. In the morning, FLW would spend an hour or two completing the rendering, ready for a lunch-time meeting with his clients. Today, many architects use their computers in the same way as FLW used his night staff, to set-up an outline perspective, over which a rendering is produced by hand. Students, in particular, can often be observed attempting to complete a basic rendering by using a paint program such as Photoshop to apply textures and entourage in a kind of electronic collage.

Photorealism has been espoused by computer graphics researchers because of its easy determinacy. If you specify all the light sources, the optical characteristics of all the materials, and simulate the physics of light with the utmost fidelity, then the resulting image is fully determined; there is nothing left to the imagination, and it is the ideal subject for a computer program of the old-fashioned kind. But this hardly meets the needs of a visually oriented profession, which is accustomed to using images day in and day out, and sweating the maximum benefit from each one. Imagination has to be let back in.

The photorealistic image is problematical in practice. It tends to be read as a complete, definitive statement. It is non-negotiable. Sometimes this is appropriate, but there are many times when something fuzzier, more tentative is called for, either because ideas are only partially worked out, or to encourage discussion. In our work images can contain any amount of "noise" which people will tend to interpret according to their own predilections. As Gombrich(1960) tells us, such images stimulate the imagination, rather than suppress it; and as an added benefit, require less geometrical detail in the original CAD model, and so are both economical and especially appropriate at the early stages.

Photorealistic practice ignores the *selectivity* which is fundamental to all kinds of freehand drawing. Everything is rendered in the same way, with the same amount of detail. Piranesi offers infinite ways of foregrounding the subject, by modifying its weight, colour or textural detail, and reducing background by fog, blur or vignetting; and this hardly requires conscious effort. The way we judge these renderings has shifted – we are no longer talking about technical virtues such as the smoothness of shading, the sharpness of textures, or the size of highlights, but are concerned with their pictorial qualities, of how they focus an issue and stimulate the imagination or support an argument.

1.2 Rendering in practice

Computer rendering of architectural proposals is becoming increasingly common, and increasingly specialised, usually being carried out by an expert group within the office, or very often by an external firm. The software employed for the most advanced photo-realistic images is often complex, slow, expensive and demanding.

If we look at the workflow seen in a commercial rendering house, it moves through recognisable stages. Site photography is commissioned and scanned to provide a background. The proposal is modelled in 3d, specifically for visualisation, and only to the extent needed to achieve the required views. It is then rendered, often in several layers such as background, elevation, and as much interior as is visible through windows. Then these layers are combined with the site photography in

Photoshop, entourage added, and the whole thing finished by manual editing. Each stage may take several days of skilled operator time to perform.

We are particularly interested in the last two stages, that of rendering and photo-finishing. Rendering is far from interactive - it includes placing lights and textures, adjusting many parameters, and carrying out many trials. By contrast the photo-finishing stage is done by direct manipulation; this is a much better environment for the artist, and it is here that the visual balance of the final image is achieved.

2. THE PIRANESI PROJECT

2.1 Interactive rendering

The Piranesi project at the Martin Centre was set up to address the two issues of allowing more appropriate style of rendering than the photorealistic, and finding a mode of interaction that is a better fit to the cognitive style that visually creative people employ. The result is a new kind of program; it produces images of three dimensional objects comparable with traditional renderers, but with an interactive interface akin to a painting program such as Photoshop or Fractal Design Painter. It could be termed, with equal justification, a 3D paint system, or an interactive renderer.

Piranesi is a software system that succeeds in shifting the boundary between the rendering and the finishing stage, making much more of the process direct and interactive. To use it a basic rendering is made, without textures, and exported to Piranesi with its z-buffer, which gives the distance to each pixel, and a material buffer which encodes the layer or material in the model from which each is derived.

Piranesi itself is structured like a paint system, with the usual sort of brushing and bucket-fill operations. But the presence of the z-buffer gives it many new capabilities. By using the z coordinate to modulate painting effects, it becomes sensitive to depth. By differentiating the z-buffer, it can reconstruct the surface normal, and so become sensitive to orientation. By differentiating twice it can recover curvature, and detect physical edges. Finally, by combining pixel position with depth, and passing it through an inverse perspective transform, Piranesi can reconstruct the original 3d coordinate at any point. Some of the more interesting effects enabled by the z-buffer are:

1. Restriction of a stroke to a particular plane, or to planes with a particular orientation. Combined with restriction based on material, this removes the need for selection masks. As the typical Photoshop user spends more time making masks than doing anything else, this is by itself a substantial advance.

2. By making the paint effect proportional to depth, we get the effect of fog or aerial perspective. By using various functions of depth we can achieve haloes around a point, or the effect of mist rising from the ground.

3. Textures can be brushed on in a conventional 2d sense. But by using the z-buffer we can map textures by projection, either frontally or by normal projection onto the surface being painted. This changes texture generation from a hard to control batch-mode process in the renderer, to an easy piece of direct manipulation. Textures can be applied experimentally and built up by layers.

4. By relating paint effect to the surface normal, we can achieve lighting effects outside the renderer.

5. Scanned images can be pasted in at a definite depth in the image. So they scale according to the perspective, and can obscure and be obscured properly. They can also be fixed onto surfaces, giving the effect of a poster or a creeper growing up the wall.

6. Edge detection in the z-buffer leads to a peculiarly sensitive kind of line-drawing. Silhouette edges are naturally emphasised over internal edges, while shadow edges do not register at all.

Piranesi is constructed as a re-rendering system rather than (as Photoshop) an editor of a single image. The original image is retained through the process, and is sampled in various ways as the new image is painted. So a texture may provide its own colours, or a constant colour may be used, or the original image may be sampled to provide the colour. Sometimes the output image is initalised to the input image, but on other occasions we may prefer to start with a monochrome screen.

Piranesi can achieve most of the effects of a conventional photorealistic renderer, but is even more succesful with non-photorealistic rendering. It is not governed by the laws of optical physics, but by the eye and judgement of its user. This vastly increases its creative potential.

Non-photorealistic rendering will typically proceed in two passes. In the first, colour and shading are added to the original image, plus entourage and background, defining the composition and overall tonal balance. This becomes input to a second pass which starts with a blank sheet. Edges are pulled through, like a pencil outline. Then textures which sample the input colours are applied. These may be 2d or 3d procedural textures, or vey often hand-drawn and scanned samples, which are mapped into perspective by Piranesi. A technique akin to half-toning can produce effects like pen and ink drawing, or wood-engraving. The amount of detail, and hence the density of such a texture, is controlled by the tonality of the input image.

2.2 Related work

Some of the fundamental ideas of Interactive Rendering (in particular the separate treatment of geometry and appearance, and the enriched pixel) were suggested by Perlin's Pixel Stream Editor of fourteen years ago (Perlin, 1985). However, it was far from interactive, and required that each pixel should be processed independently, which prohibits many desirable operations. The ideas of fractal noise, turbulence and solid texture included in the same paper have been immensely influential.

Perlin was rather vague about what data should be associated with each pixel; Saito and Takahashi (1990) by contrast define a comprehensive G-buffer that contains identifier, parametric coordinates, world coordinates, depth and direction cosines. This is used to support perspective space hatching and edge finding, with the idea of making technical illustrations "comprehensible". We differ in having some interest in ambiguity as well as clarity, and preferring an interactive process to one that batch treats a complete image. We have found it possible to reduce the G-buffer to a pair of numbers, without serious loss of effect.

The idea of image space rendering using synthetic paint marks was important in the genesis of our project. Similar ideas have been published by Haeberli (1990), and have found their way into commercial products such as Fractal Design Painter. It was about this time that the term non-photorealism came into use. Three-dimensional painting by means of reverse texture mapping was first described by Haeberli & Hanrahan (1990). It is beginning to appear in commercial products (such as Painter 3d), but requires high performance hardware to be interactive. It works purely in perspective space.

A group at the University of Magdeburg has been looking at computer-generated ‘sketches’ and, more recently, at rendering techniques for medical illustration, another discipline where photorealism is less than ideal(Strothotte & Strothotte 1997). The work on pen and ink drawing simulated by stroke textures by Winkenbach & Salesin (1994, 1996) and Salisbury, Wong, Hughes & Salesin (1997) at the University of Washington is closer in spirit to ours, and has been applied specifically to architectural rendering. Originally a batch mode renderer, recent papers indicate a growing interest in making the process interactive. Some of the latter work has been commercialised by Inklination Inc, while the former has resemblances to the LiveArt 98 software produced by Viewpoint Data Labs International Inc.

The modern world of digital image-making has been comprehensively surveyed by Mitchell (1993). Early work on this project has been described by Richens (1994, 1995). A comprehensive review of non-photorealism can be found in Schofield and Lansdown (1995), and additional discussion in Strothotte & Strothotte(1997) . There will be a workshop on the subject at SIGGRAPH 1999.

3. EXAMPLE

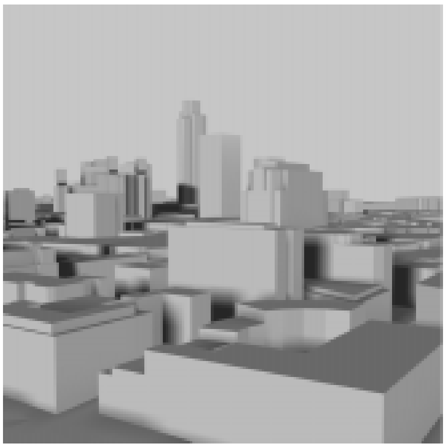



Figure 1,2

The input to Piranesi is a kind of extended TIFF file, containing an initial rendering (Fig 1) and the z-buffer (Fig 2). The latter encodes, for each pixel, how far away the object rendered is. So near objects are dark, distant ones light. It also contains a material index for each pixel.

The rendering we are using here was made by 3D Studio Max, using a freeware plug-in renderer to export the interface file. The scene was generated by the London architects Miller+Hare, from their urban model of the city. Most of the buildings are simply extruded from their footprints on a map, though a few of the more prominent have been modelled in more detail. The model is simply lit and untextured, which is of no consequence, as both will be reconstructed in Piranesi.

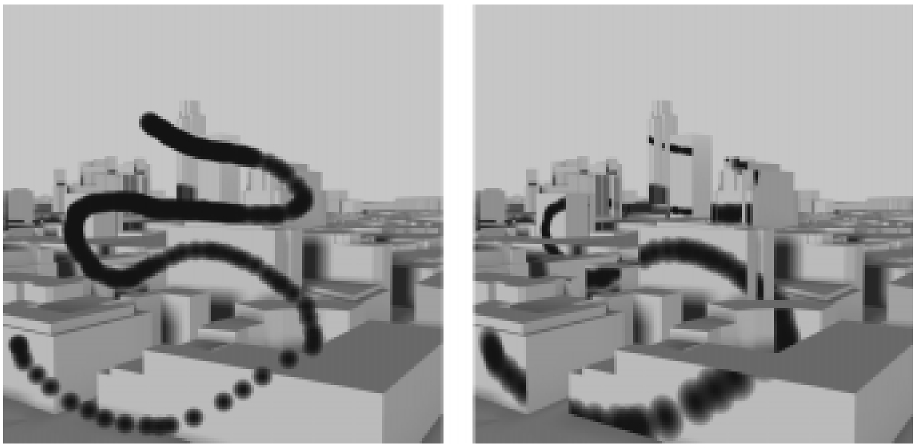


Figure 3,4

The z-buffer is what makes Piranesi unique among painting systems. The first consequence, and perhaps most startling to a Photoshop user, is that is does not need selection tools and masks to limit painting to particular areas. Instead it uses *Locks* to constrain painting either by material, or by geometry. The most commonly used is *Plane Lock*, which constrains paint to pixels which lie in the same plane as that at the start of the stroke. (The relevant plane equation is determined by calculating the gradient of the Z-buffer). Here (Figure 3) we show an unconstrained stroke, compared (Figure 4) with one made with *Orientation Lock*. This selects the planes facing in the same direction as the one where the stroke started. In Fig 3 we use a standard 2d brush, but in Fig 4 the stroke is repeated in 3d. So the brush outline becomes elliptical, and its size diminishes as it gets further away.

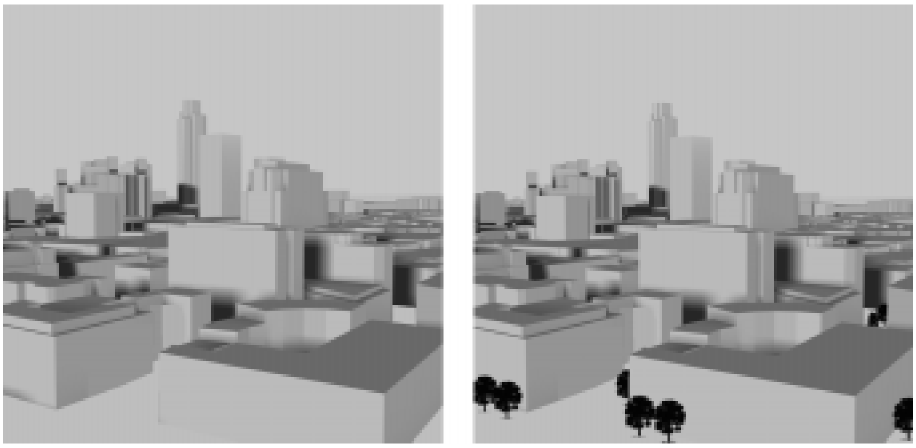
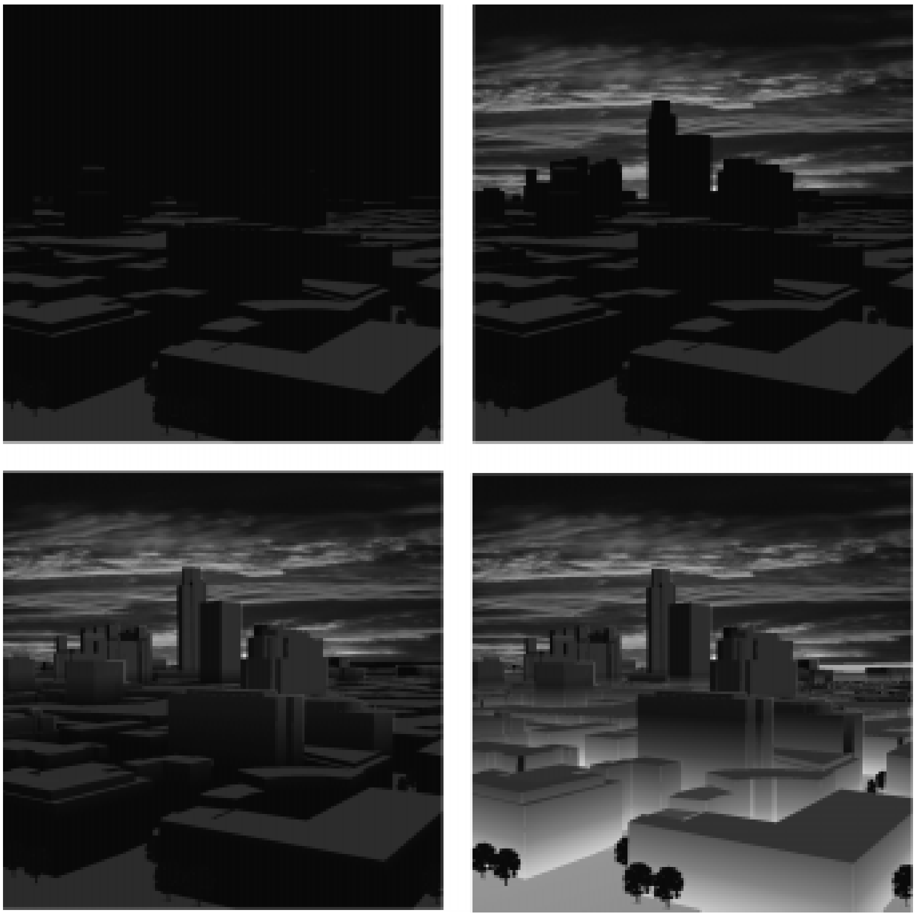


Figure 5,6

The first step in a Piranesi rendering is usually to correct small modelling errors using a tool which operates on the z-buffer by extending existing planes. Here

(Figure 2) it is used to modify the foreground building so that the street behind is more visible. We are not concerned with realism, but with making a picture.We need to be able to see into the street, so we demolish some building.

In Figure 6, we paste in a few cut-out rees, to establish the scale. Pasting in Piranesi is in perspective, true to scale (by default), and deals with obscuration. So trees know what they obscure, and what obscures them, and the distant ones are smaller than those nearby, according to the perspective of the scene.



Figures 7,8,9,10

We now start to paint the lighting, with the intention of capturing the general effect of a city just after sunset, when the electric lights come on. Starting with a black screen (Fig 7), we allow a little light onto the rooftops, by painting with *Orientation Lock*. Then, selecting a sky shot as a texture, we paint in the background (Fig 8). The high buildings will catch the last of the sunset glow (Figure 9), so we paint them with two grading effects. One keeps the light stronger as it is higher, and the other takes the plane orientation into account, so giving the effect of directional lighting from the back right. Finally we add an overall approximation to the artificial light in the scene, which is generated at street level, and fades out with height. This light, in reality the result of thousands of indivdual sources, is approximated by another vertically graded fill.


Figure 11,12

In Fig 11 we start to paint windows into the scene, by using very simple black and white texture tiles representing various fenestration patterns. The actual colour used is decided when the paint is applied; here it is sometimes light, sometimes dark. As the textures are brushed on, this sort of detail variation is perfectly natural. Piranesi automatically puts the texture into corrrect perspective according to the depth and recession of the plane on which it is being painted.

In Piranesi it is very easy to build up layered effects. For example the figure of a model in Fig 12 was pasted onto the face of the tall building, but transparently. This operation changed the material buffer only. Next she was textured with a pattern of lights, using the material lock, so picking out the silhouette.

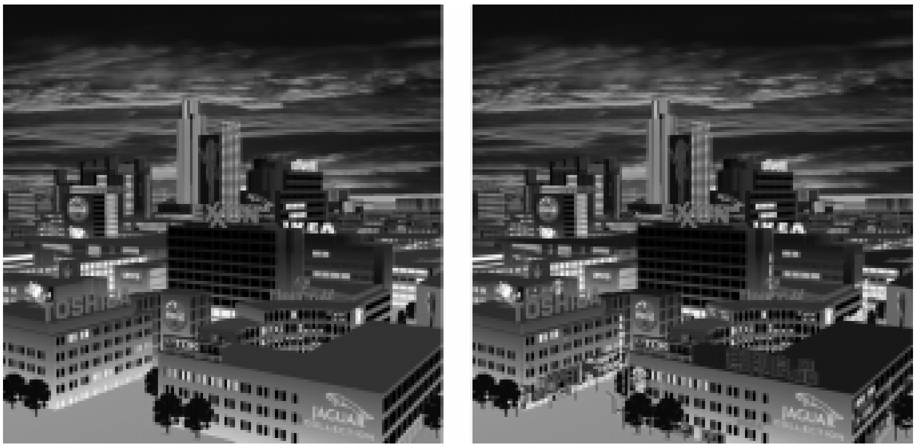


Figure 13,14

Neon signs are added in much the same way as trees were at the beginning. In figure 13 rooftop signs are generally aligned with a wall, and then slid into place on the roof. These cut-out elements may be fully coloured, or simply greyscale, in which case the colour is provided at the time of insertion. When one is inserted, the z-buffer values are inspected. Only when the insert's z value is less than the existing, is it painted in. Where it is painted, the z and material buffers are also updated. This ensures that future cut-outs will correctly obscure or be obscured by the first one, and that it may be repainted or textured subsequently to its insertion.

The use of cut-outs saves a huge amount of modelling effort in dealing with details such as trees,people, railings and fences, or, as in this case, neon roof signs. In Figure 14 street lamps have been added in the same way, and a lot more signage at street level. This is usually applied flat onto the wall, but for little more effort can be rotated to project out from it.

Figure 15,16

A first approximation to lighting was painted in earlier, before it was motivated by any apparent sources. Now it is refined by painting a glow around individual sources. The pools of light beneath street lamps are done using a soft-edge circular brush. Because Piranesi works in perspective, the brush marks are rendered by perspective into appropriate ellipses. Glows can also be arranged by defining a source position, and then using a painting effect that is graduated by distance from that source. Piranesi does actually include algorithms equivalent to those for conventional computer graphics point (omni) and directional light sources, but everything here is mediated by the brush.

Another way of providing glows is to use a blur filter. In Fig 16 the horizon glow as been arrived at by using some distance-dependent mist, plus some distance-dependent blurring.

Figure17

This concludes the development of the lighting, but pictorially there is something wrong – there is no foreground. This is quite readily rectified by constructing a girder pattern from a greyscale texture. Again this is located in perspective, and so is really there in the z-buffer. The edge tool can find it, and as a finishing touch the edges are allowed to pick up a little reflected light, which instantly gives the flat texture a three-dimensional feel (Fig 17).

3.1 Re-rendering

So far, everything has been done using the original Studio Max render as input. But it is possible to go further, using the new rendering of Figure 17 as input to further stages. As the work so far has been done in colour, and this paper will probably be published in black and white, some of the techniques in Piranesi for producing non-photorealistic black and white graphics will be explored.

Figure 18

The first example (Fig 18) is a binary bit map, highly suitable for output on a laser printer. It is generated using a hand-draw pen and ink cross hatch texture, which is capable of responding to the darkness in the input image by varying the thickness and density of the hatch lines it produces. This technique is called thresholding, and can produce a number of duotone image effect, including etching and engraving. Here the image retains the characteristics of pen and ink sketch. The texture has been applied in a 2d sense in the background of the image, but in the foreground it has been texture-mapped onto individual planes of the buildings, so that the hatch lines respond to the perspective. This ability to work in both perspective space and on the surface of the paper, in any combination, is a unique feature of Piranesi.

Figure 19

Figure 19 is another example of using a texture, in this case a computer generated screen texture of slightly wavy horizontal lines. This time it is applied with a softer sigmoid thresholding function rather than a sharp step. It gives a result quite similar in its chiaroscuro to a nineteenth century wood engraving. The sky was rendered first, using the texture in 2d, at a fairly large scale. Then the foreground was painted plane by plane, using the texture in perspective. The background cityscape was finished using the texture rotated ninety degrees, and at a smaller scale than was used for the sky, but similarly in 2d.

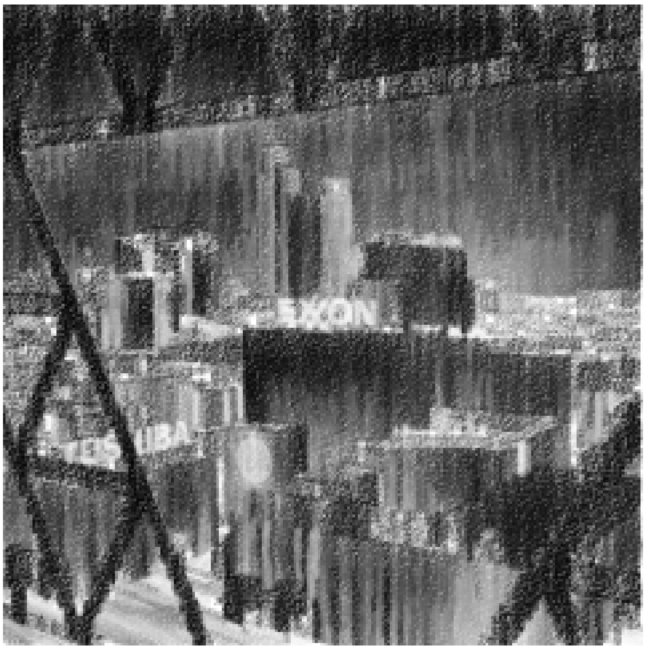


Figure 20

Figure 20 is generated by quite a different technique. First a grain texture was selected, with a surface similar to water-colour paper. This texture is used in 2d, over the whole surface of the picture, and basically affects how readily the output image becomes coloured. The rendering was done using automatically generated marks, vertical lines with a slight random wobble, whose colour is generated by sampling the input image at the middle of each stroke. The marks themselves are clipped to plane boundaries. The resulting image has low definition, and is full of noise. There is a definite resemblance to charcoal or graphite stick drawing. As one would do when using these media, the highlights (such as the neon signs) are picked out afterwards using an eraser, constrained using the *material lock*.

Figure 21

The final impression of this scene (Fig 21) is rendered in a manner analogous to mezzotint, and has a similar effect. In mezzotint, the copperplate is roughened all over until it prints a perfect black (a very lengthy process), and then the image is formed by burnishing away the roughness to draw out the light tones. In Piranesi you start by filling the screen with black, and then select an appropriate grain – in this case a fractal noise function was used. A first pass is done automatically, at very low intensity, so that you can see the ghost of the image. Then, keeping the intensity quite low, the lights are brushed over repeatedly, gradually building them up to strength. This is best done using a tablet and pressure-sensitive stylus. Because of the low-intensity setting, and the use of the grain noise, it takes many passes to bring a light up to the strength it has in the original image. This gives great control over the development of the image. The lock functions can be used, but are not so necessary in this sort of slow image making. Here we used material lock at first, to develop the sky independently of the buildings. Towards the end the edge tool was used a little, to pick out silhouettes in white.

4. CONCLUSION

Using Piranesi has a profound effect on how you go about making images. The difficult stage of setting material parameters and texture positioning in a batch-mode renderer is eliminated. Instead the image is directly constructed, in a surprisingly short time (typically 30 minutes). For the first time, it is possible for the rendering artist to sit down with his client architect or developer, and try out 3 or 4 presentational styles before his eyes. This brings computer presentation back in line with the traditional architectural approach of talking while drawing, in a very satisfactory way. The software itself is no more complex than Photoshop, and needs similar hardware resources and operator skills.

Piranesi uses the computer as a medium - not a simulation of a traditional medium, but something unique and new, a three-dimensional paint-box. Behaving as a medium is quite different from behaving as a heavy duty analytical engine, which is what the normal photorealistic renderer is. The fundamental difference is that there is an immediacy to the result, you see the effect not a few minutes or a few seconds after you make a move, but continuously, within milliseconds. This feeling, of a complete connection between what you are doing and what you are seeing, is even stronger if you use a tablet with a pressure sensitive stylus. You now focus entirely on the image, rather than on the process of making it, and the work “flows”. This is the cognitive shift to “right brain mode” described by Betty Edwards(1982). With traditional paint programs this flow keeps breaking down, because you have to make selection masks for nearly every operation. Piranesi uses *Locks*, and does not even have a selection tool. It also avoids any kind of numeric input (sliders or direct manipulation instead), and is very close to being keyboard-free. The consequence is that Piranesi is a fast, expressive, absorbing, and uniquely enjoyable way of rendering images on a computer.

5. ACKNOWLEDGEMENTS

We acknowledge with gratitude the initial inspiration for this work, which was the talk by van Bakergem and Obata (1992) at the CAAD Futures conference in Zurich, on “wiggly pen” plotting, which demonstrated conclusively that computers, just like draughtsmen, need to “loosen up” before they can produce interesting drawings.

Piranesi was developed in prototype form at the Martin Centre CADLAB by Simon Schofield (an artist), with support from Informatix Inc. A commercial version was released by Informatix Software International in January 1998. We are particularly indebted to Masanori Nagashima, President of Informatix, for his long-term support, and to Brian Woodward of Informatix Software International, who led the implementation team. A number of the techniques described in this paper will not be available until the release of the second version, expected in the fall of 1999. Part of the software is protected by US Patent Application 4,558,302, and equivalents is other countries. For a downloadable demo and details of the commercial version please see the Informatix web site at www.informatix.co.uk.

6. BIBLIOGRAPHY

Bakergem, W.D. van and Obata, G. (1992). *Free-hand plotting - Is it live or is it digital?* In (ed) G.N. Schmitt, CAAD Futures. Vieweg, Wiesbaden.

Edwards B. (1982) *Drawing on the Right Side of the Brain*. Fontana, London, p57.

Gombrich, E.H. (1960). *Art and Illusion: A study in the psychology of pictorial representation*. Phaidon, London.

Haeberli, P. (1990). Painting by numbers: abstract image representation, Computer Graphics, vol. 24, no. 4, ACM SIGGRAPH, pp. 207-214.

Haeberli, P. & Hanrahan, P. (1990). *Direct WYSIWYG painting and texturing of 3-D shapes*, Computer Graphics, vol. 24, no. 4, ACM SIGGRAPH, pp. 215-224.

Mitchell, W.T. (1993). *The Reconfigured Eye*. MIT Press, Cambridge MA.

Perlin, K. (1985). *An image synthesiser*, Computer Graphics, vol. 19, no. 3, ACM SIGGRAPH, pp 287-296.

Richens, P. (1994). *Does knowledge really help?* In (eds) G. Carrara and Y.E. Kalay, Knowledge-Based Computer-Aided Design, pp. 305-325. Elsevier, New York.

Richens, P. (1995). *Interactive computer rendering*, Architectural research quarterly 1:1 Autumn 1995. London, Emap press, pp82-95

Salisbury, M., Wong, M., Hughes, J., & Salesin D. (1997) *Orientable Textures for Image-Based Pen-and-Ink Illustration.* In Computer Graphics Proceedings, Annual Conference Series, 1997. ACM SIGGRAPH, pp. 401-406.

Saito, T. and Takahashi, T. (1990). *Comprehensible rendering of 3-D shapes*, Computer Graphics, vol. 24, no. 4, ACM SIGGRAPH, pp. 197-206.

Schofield, S. and Lansdown, J. (1995). *Expressive rendering: an assessment and review of non-photorealistic techniques*, Computer Graphics and Applications (May), IEEE.

Schumann, J., Strothotte T., Raab A., Laser S. (1996). *Assessing the effect of non-photorealistic rendered images in computer-aided design*, in Proc. CHI '96, Human Factors in Computer Systems, Vancouver BC, pp 35-41.

Strothotte C. & Strothotte T. (1997) *Seeing between the pixels*. Berlin, Springer-Verlag.

Watts, A. (1962) *The Way of Zen*. Penguin Books pp214-5

Winkenbach, G. & Salesin, D.H. (1994). *Computer generated pen-and-ink illustration.* In Computer Graphics, Proceedings, Annual Conference Series, 1994. ACM SIGGRAPH, pp. 91-100.

Winkenbach, G. & Salesin, D.H. (1996). *Rendering free-form surfaces in pen and ink.* In Computer Graphics Proceedings, Annual Conference Series, 1996. ACM SIGGRAPH, pp. 469-476.