EG UK Theory and Practice of Computer Graphics (2006), pp. 1–10
M. McDerby, L. Lever (Editors)

State of the Art Non-Photorealistic Rendering (NPR) Techniques

Rezwan Sayeed and Toby Howard

School of Computer Science, The University of Manchester

Abstract

The emergence of non-photorealistic rendering (NPR) over the greater part of a decade has created an intriguing new field espousing expression, abstraction and stylisation in preference to the traditional computer graphics concerns for photorealism. By lifting the burden of realism, NPR is capable of engaging with users, providing compelling and unique experiences through devices such as abstraction and stylisation. Many artistic and visual styles have been achieved by NPR including interactive and automated systems for drawing and painting. In this paper we outline the current state-of-the-art of NPR for visualisation and identify some current and future trends in NPR research.

Categories and Subject Descriptors (according to ACM CCS): I.3.0 [Computer Graphics]: General; I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.6 [Computer Graphics]: Methodology and Techniques

1. Introduction

Non-photorealistic rendering has two complimentary goals: the communication of information using images; and rendering images in interesting and novel visual styles which are free of the traditional computer graphics constraint of producing images which are "life-like".

Foley et al. [FAFH90] note that in achieving comprehensibility, an image free of the complications of shadows and reflections may well be more successful than the most advanced photorealistic image. In optimising an image for comprehension, by abstracting away extraneous detail and representing detail using an appropriate stylisation to create a 'perceptually-efficient image' [GIHL00], it is possible to achieve this aim. Early work in non-photorealistic rendering was largely concerned the reproduction of purely aesthetic human artistic styles [LS95]. In this paper we seek to outline the current state of technical visualisation in NPR research.

This paper is organised as follows. Section 2 details NPR lighting models. Section 3 examines silhouette and edge rendering methods that form the core of many NPR methods. Section 4 details illustration methods using only pen-and-ink styles and their derivatives, and Section 5 follows the recent application of NPR techniques to volume rendering. Formal evaluation of perception in NPR methods is examined in Section 6, and finally we present a conclusion in Section 7. Note: because editorial limitations prevent us from including many example images in high-quality reproduction, we have created a companion website to this paper, at http://www2.cs.man.ac.uk/~sayeedr2/npr/, where the reader may find a gallery of images illustrating the research surveyed in this paper.

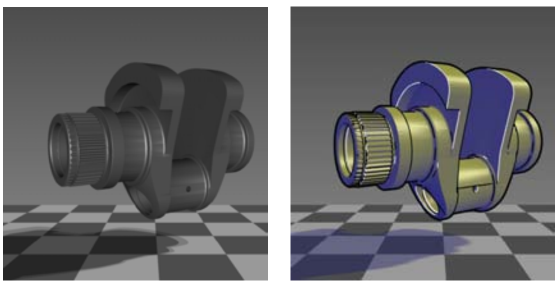

Figure 1: *Lighting enhancement: Phong shading (left) and tone shading enhancement (right) based on warm and cool colours with bold silhouettes [GGSC98, GSG*99]. Used with permission from Bruce Gooch.*

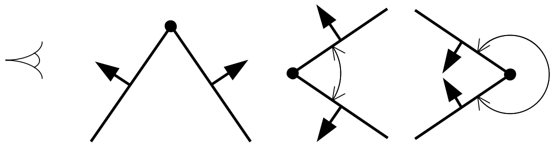

Figure 2: *Edge detection. From left to right: Silhouette edges between front and back-facing polygons, ridge and valley feature edges by dihedral angles.*

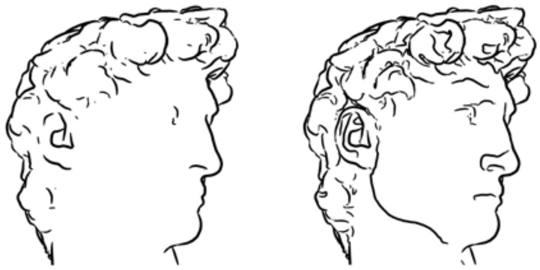

Figure 3: *Illustration enhancement: Additional classification of feature edges improves comprehension [DFRS03]. Used with permission from Doug DeCarlo.*

2. Stylised Lighting

2.1. Tone Shading

Photorealistic lighting has many complications for use in general technical illustration. Lighting may not always adequately delineate objects and details, and shadowing may obscure surface detail and shape. Gooch et al. [GGSC98] pioneered a lighting model simulating the cool-to-warm shades found in technical illustration. This is achieved by clamping the range of colours used in shading to predefined cool and warm colours, k_{cool} and k_{warm}, with the dot product of the surface normal (n) and light source vector (l) used to differentiate shades. This remains the standard method of shading with applications ranging from technical illustration [GSG*99] to volume rendered illustrations [ER00]. An example illustration is found in Figure 1.

2.2. Cartoon Rendering

Cartoon lighting is a distinctive style characterised by large uniformly or widely banded coloured surfaces with detail largely restricted to outlines. Decaudin [Dec96] introduce a cartoon stylisation model making extensive use of G-Buffers from [ST90]. The diffuse reflection $n \cdot l$ is removed from the Phong lighting model but specular highlights are retained for information about surface type. Lake et al. [LMHB00] conversely threshold $n \cdot l$ with 0.5 on a per vertex basis to render regions of uniform colour. By changing the threshold and assigning bins, it is also possible to have multiple uniquely coloured regions.

Many models for approximating realistic specular highlights exist, but it is often desirable to exaggerate or stylise specular highlights to achieve bold results in technical illustration or animation. Anjyo and Hiramitsu [iAH03] demonstrate a stylised specular highlight model for animation. After initially constructing a specular highlight, their method proceeds to apply a series of transformations including scaling, rotation, splitting and squaring to achieve effects such as striped highlights on shiny surfaces.

3. Silhouettes and Edges

Line drawings constitute one of the most common and effective illustration styles. Constructed purely from monotone line strokes, examples are found in formal contexts such as technical illustration, architectural design and scientific diagrams, and artistic contexts including cartoons and sketches. Line art techniques can express information through an image in a concise and often abstract manner.

In tracing only the outline of an object, a *silhouette*, it is possible, using very few strokes, to delineate objects in a scene and communicate facts including object size and shape. Silhouettes may be rendered in various styles to influence perception [SSLR96] or for aesthetic value in mimicking human artists. Object outlines can also be integrated with photorealistic images [RTF*04] or formally shaded images [GGSC98] to enhance comprehensibility.

We now describe some classifications used in edge rendering, followed by an examination of object space, image space and hybrid rendering approaches.

3.1. Edge Classification

In general for a surface S, its silhouette is the set of all points on S where $n_p \cdot (p - c) = 0$, given surface normal n_p at point p and camera origin c (which of course implies that silhouettes are viewpoint-dependent). The lack of normals for arbitrary points of a polygonal surface leads to a different approach. The silhouette edge set of a polygonal model is defined as the set of all edges shared by both a front and back facing polygon, the change of sign in the dot product indicating a silhouette edge, as shown in Figure 2.

The detection of useful edges is not limited to silhouettes — additional information can be implied by rendering particular edges. A *crease* is an edge shared by two front-facing polygons where dihedral angles are compared to a threshold. Convex and concave features, ridge and valley edges, have dihedral angle thresholds either side of 180°—120°and 240°for example [Ras01] (see Figure 2). *Border* edges appear in open polyhedral edges and have only one adjacent face. DeCarlo et al. [DFRS03] extend classification of surface features through 'suggestive contours', providing a formalised definition of interesting contours based on the radial curvature characteristics of the surface (see Figure 3). Sousa et al. [SFWS03] use geomorphological measures such as slope steepness and aspect in classifying useful feature edges for pen-and-ink illustrations.

3.2. Object Space Methods

Object space silhouette rendering methods operate directly on the underlying 3D geometry of a model. This information permits direct comparison of surface normals and provides data for stylised renditions of silhouettes edges. Using this information, Hertzmann and Zorin [HZ00] interpolate between edge vertices with dot products of differing sign to construct a piecewise silhouette of a high quality.

The naïve approach to silhouette extraction involves testing each edge in the model to verify whether it is a silhouette for each frame of animation. Additional edge adjacency information is also required for dot product calculations. Although simple to implement, robust and fast for low detail models [KGC00], brute force methods do not scale well for detailed models, where the key challenges are accurate and fast detection and visibility determination of edges.

3.2.1. Probabilistic Techniques

Markosian et al. [MKT*97] provide a search heuristic for initially finding a silhouette edge and then use the spatial coherence of silhouettes to rapidly trace out an entire silhouette outline using adjacency information. The initial search requires examination of a constant $O(\sqrt{n})$ number of edges, further improved by observing that the probability of a silhouette edge is inversely proportional to dihedral angle. An enhanced version of Appel's hidden line algorithm [App67] is used for visibility calculations.

For accelerated inter-frame calculation of silhouettes, a small fraction of silhouette edges are retained and used as the basis for subsequent searches. DeCarlo et al. [DFR04] observe a detection rate of almost 80% using this method.

This approach offers a significant performance improvement over a brute force search of all silhouette edges. However, silhouettes in regions unconnected to the initial search may not be detected.

3.2.2. Edge Buffer

The edge buffer, introduced by Buchanan and Sousa [BS00], provides a table data structure that enables detection of silhouettes and borders in an object. The structure stores an indexed list of vertices with attributes including adjacency information and front and back-facing flags, F and B.

The FB flags used in the table can be used to determine whether an edge is a silhouette. A pre-process is needed where the table is initialised on a per-polygon basis during rendering. The relevant front/back facing flag is inverted depending on the orientation of the polygon using a XOR operation. When complete, edges with $FB \neq 0$ are rendered as silhouettes.

3.2.3. Hardware Geometry Modification

Raskar [Ras01] describes a method of introducing geometry to represent silhouette and crease edges. This occurs at the primitive shader stage, where single polygons are processed at a time, leading to efficient determination of front and back facing polygons.

Applying this approach enables silhouettes, ridges and valleys to all be detected using the same technique. Using primitive assembly shaders, uniform-width quadrilaterals independent of depth are introduced at the feature edge and orientated towards the camera. All quads are coloured in black. This permits robust rendering, but missing end cap artefacts are found when rendering wide silhouettes on surfaces with low tessellation and high curvature.

The rendering of stylised quads at feature edges can also be implemented and accelerated on the GPU through vertex programs. Due to the inability of the current generation of graphics hardware to create and delete vertices in the vertex processor, vertex lists are augmented with additional degenerate vertices that are rendered only if a silhouette is detected. If not found to be visible, the quad is culled by translating its vertices behind the near clipping plane.

This *degenerate* geometry idea is used by Lengyel et al. [LPFH01] under the term 'fins' in rendering fur outlines of objects. Card [CM02] provides a hardware implementation of degenerate geometry, packing adjacent face normals into vertex attributes and storing 4 vertices per edge into the vertex buffer. McGuire and Hughes [MH04] follow this general approach but store adjacent triangle vertices instead of explicit surface normals, allowing introduction of end caps to partially solve the issues from [Ras01], though incur further vertex buffer overheads.

3.2.4. Frame Coherence

The 'stability' of a silhouette refers to the likelihood of a silhouette being visible following small changes in view. Unstable silhouettes appear and disappear quickly, resulting in a distracting effect for viewers. Lengyel et. al. [LPFH01] provide a scheme for using the dot product of the surface normal with the view vector to vary the opacity of the quad to 'fade in' and 'fade out' rendered quads to achieve improved frame coherence. Brosz et al. [BSS04] define a classification scheme for edges and corresponding opacities. DeCarlo et al. [DFR04] apply stability to suggestive contours, using motion measures to discard contours that move too fast across a surface and using opacity to fade stable remaining suggestive contours.

3.2.5. Hierarchies

Gauss maps may be extended to provide visibility tests for faces [ZKEH97] and silhouettes [BE99, GSG*99]. A sphere is placed around the object and regions of the sphere from which a silhouette is visible are computed and stored. The sphere may be mapped to a bounding cube and surfaces further subdivided using bin methods such as octrees. BSP visibility methods may also be applied. Testing for silhouette visibility is hence reduced to searching the hierarchies on these planes, achieving performance logarithmic in the number of edges. Gauss maps are ideally suited to orthographic projections but require additional searching under perspective projections.

Hertzmann and Zorin [HZ00] use the notion of a dual surface to detect silhouettes. A duality map transforms surfaces into dual space. Silhouette detection is reduced to the intersection of the viewplane with the dual surface, yielding a set of silhouette points. Acceleration using hierarchical visibility methods is possible. A similar approach is employed by Pop et al. [PDB*01], deriving silhouettes by intersecting point (or edge) duals with the viewpoint dual plane.

Sander et al. [SGG*00] create a hierarchy of mesh edges and faces as a pre-process, where each subtree is either a cluster of front-facing edges or back-facing edges. Anchored nodes are stored at each node indicating viewpoints for which clusters are either all front or back-facing, permitting constant time visibility testing.

3.3. Image Space Methods

Image space silhouette rendering methods lack the abstraction afforded by geometrical primitives and hence follow a very different approach, using image processing algorithms. Although image space techniques are generally fast and relatively detail-independent (resolution being the greatest constraint), the lack of geometry information reduces the variety of line styles achievable.

3.3.1. Discontinuities

Saito and Takahashi [ST90] compute discontinuities in the depth buffer using a first order differential operator and aggregate the result with the rendered image to enhance comprehensibility. They introduce the notion of a *geometric buffer* or *G-Buffer* that stores two-dimensional data. These are used to accumulate intermediate results for a final image, including features such as depth, silhouettes, hatching and shading.

Mitchell demonstrated the execution of these methods in hardware [MBC02], extending the source of discontinuities to the normal, depth and id buffers using fragment shading hardware. Nienhaus et al. [ND03] in addition use the normal buffer to detect crease edges, and together with silhouettes, project the resulting G-Buffer onto object space.

3.4. Hybrid Approaches

Hybrid silhouette rendering techniques benefit from the speed and accuracy of image space silhouette detection with stylisation capabilities afforded by access to underlying 3D geometry.

3.4.1. Object-space Silhouette Extraction

Gooch et al. [GSG*99] use an environment map to illustrate silhouette edges by darkening vertices that are below a low dot product threshold. They achieve a stylistic effect with silhouette width varying with curvature.

Northrup and Markosian [NM00] present a hybrid approach for rendering artistic silhouettes. Detection and rendering is performed using the probabilistic method of Markosian et al. [MKT*97]. Then using an image, edges are detected and interpolated to form line segments and finally stylised using a variety of methods [HLW93, HL94].

Kalnins et al. [KDMF03] present a method that uses a brush path parameterisation to represent a silhouette, achieving a similar effect. An image space local search method is used to track silhouettes between frames providing frame coherence.

3.4.2. Image-space Silhouette Extraction

Rossignac et al. [RvE92] first render to the depth buffer, then translate objects backward and render fully in wireframe mode with wider lines using the updated frame buffer. Using the depth buffer for occlusion produces silhouettes by masking the rest of the model.

This method is extended by [GSG*99, RC99] both of whom use OpenGL's glOffset() function to effect the translation. Raskar [RC99] initially renders all front-facing polygons in white before switching the depth test to 'equal to' and 'fattening' back-facing polygons by shifting corner vertices. Back-facing polygons are then rendered in black, appearing as silhouettes of required width.

Ashikhmin [Ash04] describes a hardware-accelerated technique based on stencil buffer updates to detect silhouette edges. The observation that internal edges update the stencil buffer twice is used to determine and render silhouette edges.

4. Pen-and-ink, Hatching and Engravings

Pen-and-ink is a traditional human illustration method with great historical resonance. The medium itself is very limited, comprising solely of monochrome pen strokes. While strokes are ideal for outlining and filling details, the use of only black strokes leads to difficulties in expressing lighting characteristics of a surface. This is solved through *hatching*, which is a technique that indicates curvature through stroke direction and surface texture, and lighting by stroke weight, density and rotated layering (also known as cross-hatching).

4.1. Textures

Salisbury et al. [SABS94] initially introduced the notion of a *stroke texture* — an arrangement of multiple strokes in a particular pattern — as a tool to interactively generate pen-and-ink illustrations. Their method is based on a half-toning approach; a reference image is used to select a stroke texture with a matching tone. Winkenbach and Salesin [WS94] extended this work further by using *prioritised stroke textures*, a method of procedurally generating stroke textures by adding higher priority strokes until the desired tone is reached, achieving resolution independence.

A sequence of pre-computed hatching textures differentiated by tone is used by Praun et al. [PHWF01] to achieve hatching in real-time. The sequence, with corresponding mip-maps, are based on the 'art maps' introduced by Klein et al. [KLK*00]. Multi-texturing hardware is used to select and filter between tones and mip-map levels at run-time. Orientation of hatching lines is assigned according to curvature using a lapped texture technique [PHWF01]. A subsequent work [WPFH02] provided more finely-grained tone control through two schemes; volume texturing and texture thresholding, with the former supporting colour hatching.

Freudenberg [FMS02] uses half-toning threshold functions implemented in texturing shading hardware to demonstrate a range of half-toning effects and the selection of stroke textures using lighting as a target value. In earlier work, Lake et al. [LMHB00] use $n \cdot l$ to select tonal stroke textures using a bin threshold. The chosen texture is then projected onto the model directly, preserving a hand-drawn feel, but resulting in a 'shower door effect'.

4.2. Direction Fields

A key property of a stroke is the use of orientation to indicate the shape characteristics of an object. A coherent arrangement of directed lines is essential in communicating subtle undulations in a surface's shape not possible using lighting and shadowing methods that obscure detail.

Uniformly-spaced parallel hatching lines are used by Leister [Lei94] with ray-tracing to render copper plate style images, a distinctive form of hatching that uses lines of varying thickness. The use of ray-tracing is also demonstrated to produce striking hatched images exhibiting effects including refraction, reflection and shadows.

The use of principal curvature is the most common method of illustrating surface shape. Girshick et al. [GIHL00] cite the psychological basis and geometrical invariance characteristics as compelling arguments for their use. Interrante's illustration of shapes within volume data [Int97] is an early example of the use of principal curvature. [HZ00, RK00, ZISS04] all use curvature as a basis for drawing streamlines.

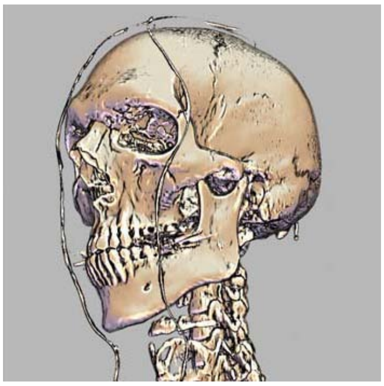

Figure 4: *Volume illustration: silhouette and crease edge enhancements with tone shading in a volume dataset [KWTM03]. Used with permission from Gordon Kindlmann.*

Parametric lines can also be used to illustrate surfaces. [Elb98, WS96] use the direction of parametric lines as a means for orientating streamlines.

Salisbury et al. [SWHS97] use interactive direction fields based on normal maps and example stroke textures to create pen-and-ink illustrations. The notion of a *difference image* was used to place strokes in a final render. The difference image is a blurred copy of the original image which is subtracted from each time strokes are placed, ensuring correct tone. Ostromoukhov [Ost99] uses patches constructed from multiple layers of dither screens to produce a variety of hatching patterns including mezzotint and various etching styles. User interaction is then used to warp patches in image space to recreate facial engravings.

5. Volume Illustration

Volume rendering has remained a prevalent tool in medical and scientific visualisation for over a decade. The ability to visualise complex real-world phenomena has found its way into practical applications including CT and MRI scans of the brain and imaging flow in fluid dynamics. The fairly recent integration of volume rendering with NPR was an intuitive and natural progression given the communicative and expressive capabilities of NPR, termed *volume illustration* by Ebert and Rheingans in 2000 [ER00].

A significant contribution of NPR is the generality of the techniques and reduced parameterisation requirements. By dispensing with the need for realistic reconstruction, and focussing on enhancing existing features without prior knowledge of voxel attributes, NPR methods provide a preliminary model for comprehension. This is important because the modulation of effective transfer functions remains a great problem, requiring expert knowledge and extensive testing with trial-and-error often the preferred option.

In the application of NPR to volumetric data sets, there are two general paradigms; feature enhancement through traditional volume rendering models, and alternative reconstruction. The reliance on volume gradients is a feature common to all volume illustration methods, used to detect features including silhouettes without geometric data.

5.1. Traditional Paradigms

Traditional paradigms in non-photorealistic volume rendering enhance features found in volumetric data in direct volume rendering (DVR) — including 3D texturing, splatting and isosurface reconstruction. Ebert and Rheingans' [ER00] initial pioneering work laid the foundation for future methods. Enhancements are achieved by increasing opacity and modifying colour at identified features, realised through additional colour and opacity modulation stages in the volume rendering pipeline. Feature detection is achieved using gradient and view direction calculations. A later work accelerated these methods using shading hardware [SE03].

Csébfalvi et al. [CMH*01] use gradient magnitude information and hierarchical voxel culling techniques to interactively display contours in a volume. Gradient magnitude is used to prioritise the appearance of detected silhouettes. Hadwiger [HBH03] adopted their shading and silhouette methods for acceleration on fragment shading hardware.

Curvature in volume rendering was used by Interrante [Int97] to illustrate surface shape using evenly-spaced streamlines directed according to principal curvature directions. Particles were used to trace streamlines over a volumetric surface. Kindlmann et al. [KWTM03] use a modified transfer function featuring a dynamic convolution-based field reconstruction method to render features based on curvature measures. Silhouettes are rendered in a similar fashion to Gooch's environment map approach using $n \cdot v$, but in addition, a curvature correction is applied to render silhouettes of fixed width in image space. Curvature magnitude is used to distinguish between ridge and valley features. An example illustration is found in Figure 4.

Lum and Ma [LM02] use hardware multi-texturing and multi-pass rendering to render tone, silhouette, gradient and depth enhancements using the contributions of individual texturing units. Rendering is completed in two passes. Paletted textures are used to arbitrarily map gradient directions to enhancements, requiring an update to the palettes upon a viewpoint change. A later work [SLM02] used an additional pass to render strokes as vector direction cues and a greater degree of pixel textures in preference to paletted textures for precision.

By sampling a 3D texture side-on, Nagy and Klein [NK04] extract one-pixel wide silhouettes by interpreting the opaque regions of texture slices as solid geometry. Then by detecting visible fragments that fail alpha tests in subsequent texture slices, silhouettes are detected. In hardware, this can be implemented in a fragment program that makes use of three texture units, corresponding to adjacent texture slices. An optional post processing stage broadens the one-pixel wide silhouette image using a Fast Fourier Transform in image space to an arbitrary width.

Markosian's randomised approach is used by Burns et al. [BKR*05] to extract contours from volume data. An initial seed cell is selected containing both an isosurface and contour is found from which a Marching Lines algorithm is applied to trace the entire contour across adjacent cells, retaining seed points across frames to improve performance. Their method is also capable of rendering suggestive contours.

5.2. Alternative Approaches

In avoiding the traditional volume rendering methods it is possible to achieve illustrations more akin to other human techniques. Treavett and Chen [TC00] demonstrate a two-phase pen-and-ink rendering procedure for volume datasets. In the first phase image buffers including shade, depth and and curvature are extracted and filtered along with the underlying sample points to achieve hatched illustrations. The ability to replace often 'medically graphic' photorealistic depictions with sanitised NPR depictions is often desirable. Nagy et al. [NSW02] construct hatched illustrations by encoding higher order differential characteristics into a hierarchical data structure and apply a prioritised stroke sampling method for run-time reconstruction.

However, surface hatching approaches fail to fully utilise the information of an underlying 3D voxel dataset. Interior data is used for hatching stroke contributions by Dong et al. [DCLK03] in illustrating medical datasets, and muscle fibres in particular. Surface hatching strokes are generated by fitting a smooth isosurface patch and then orienting strokes according to principal curvature directions. Visible interior voxels at a particular depth below the surface are rendered with strokes. Varying this depth controls the appearance of internal details.

A volume stippling framework was presented by Lu et al. [LME*02], rendering stipple illustrations from volume data using point primitives. Multi-resolution point lists are generated as a pre-process and sampled at run-time. A variety of enhancements are demonstrated, based on [ER00], and generally involve controlling the density and weight of stipples.

Although not always related to volume datasets, the use of cut-away views use principles found in volume rendering, allowing the viewer to inspect the internals of an otherwise opaque object. Diepstraten et al. [DWE03] demonstrate cutout and breakaway styles and implementation strategies. Viola et al. [VKG04] generalise the notion of cut-away views, introducing an importance-based scheme to select which parts remain visible or suppressed.

6. Perception of NPR images

Despite the fact NPR seeks to enhance the comprehensibility of images, there is no clear formal framework for the evaluation of generated NPR images. Early evaluation methods included the use of questionnaires [SSLR96] to poll architects on preferred rendering style for CAD images and the effectiveness of map layout [AS01].

Task-related evaluation methods measure the performance with which a user is able to complete a programmed task. A study by Gooch and Willemsen [GCS02] explored spatial perception in NPR virtual environments. Subjects were shown an NPR rendition of a hallway including a target, and were then asked to reach the target by walking blindly. Gooch et al. [GRG04] measure the speed and accuracy by which subjects are able to learn and recognise a set of unfamiliar faces, comparing photographs to NPR illustrations. They demonstrate that NPR illustrations can be recognised as quickly as photographs, but can be learnt from twice as quickly.

Santella and DeCarlo [SD04] proposed a quantitative framework for evaluating NPR based on eye-tracking. In a previous work [DS02], images were stylised using large regions of uniform colour and bold lines based on eye movements. A hierarchical segmentation tree was used to attenuate detail at various locations. Using clustering, regions of interest corresponding to distinct cluster groups and the number of points contained in each cluster were used to provide a measure of cumulative interest. Using these quantitative measures, they show that stylisation is unable to effect viewing patterns but abstraction is. By measuring attention patterns, it is possible to determine why and the extent to which a user is attracted by an image, rather than the accomplishment of a particular task.

7. Future Directions and Conclusion

Research in Non-Photorealistic Rendering has progressed greatly since its initial inception. The simulation of technical styles has provided a wealth of tools for illustrators, finding applications that range from illustrating mechanical parts and scientific visualisation, to medical imaging.

Future hardware trends including the creation and deletion of geometry on the GPU have wide applications for existing NPR techniques. Features such as multiple render targets and hardware-based convolution kernels have strong applications to image- based NPR techniques.

There has also been great interest in the application of stylised NPR techniques to real-world visualisation. Recent examples include the work by Raskar et al. [RTF*04] to produce NPR images by detecting feature edges using depth discontinuities and multi-flash photography. Xu and Chen [XC04] apply line stylisations to large outdoor real-world environments for the purpose of facilitating architectural design.

NPR has previously been regarded as a specialist offline or interactive-only technique. However, advances in algorithms and programmable graphics hardware, and driving forces including technical illustration and artistic expression, have allowed NPR to be realised in real-time. We should expect to see NPR techniques gain wider adoption in the near future.

References

[App67] APPEL A.: The notion of quantitative invisibility and the machine rendering of solids. In *Proceedings of the 1967 22nd national conference* (New York, NY, USA, 1967), ACM Press, pp. 387–393.

[AS01] AGRAWALA M., STOLTE C.: Rendering effective route maps: improving usability through generalization. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, pp. 241–249.

[Ash04] ASHIKHMIN M.: Image-space silhouettes for unprocessed models. In *GI '04: Proceedings of the 2004 conference on Graphics interface* (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004), Canadian Human-Computer Communications Society, pp. 195–202.

[BE99] BENICHOU F., ELBER G.: Output sensitive extraction of silhouettes from polygonal geometry. In *PG '99: Proceedings of the 7th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 1999), IEEE Computer Society, p. 60.

[BKR*05] BURNS M., KLAWE J., RUSINKIEWICZ S., FINKELSTEIN A., DECARLO D.: Line drawings from volume data. *ACM Trans. Graph. 24*, 3 (2005), 512–518.

[BS00] BUCHANAN J. W., SOUSA M. C.: The edge buffer: a data structure for easy silhouette rendering. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2000), ACM Press, pp. 39–42.

[BSS04] BROSZ J., SAMAVATI F., SOUSA M. C.: Silhouette rendering based on stability measurement. In *SCCG '04: Proceedings of the 20th spring conference on Computer graphics* (New York, NY, USA, 2004), ACM Press, pp. 157–167.

[CM02] CARD D., MITCHELL J. L.: Non-photorealistic rendering with pixel and vertex shaders. In *Direct3D ShaderX: vertex and pixel shader tips and tricks* (2002), Engel W., (Ed.), Wordware Publishing Inc., pp. 319–333.

[CMH*01] CSÉBFALVI B., MROZ L., HAUSER H., KÖNIG A., GRÖLLER E.: Fast visualization of object contours by Non-Photorealistic volume rendering. In *Proceedings of the 22nd Annual Conference of the European Association for Computer Graphics (EUROGRAPHICS-01)* (Oxford, UK, Sept. 4–7 2001), Chalmers A., Rhyne

T.-M., (Eds.), vol. 20, 3 of *Computer Graphics Forum*, Blackwell Publishers, pp. 452–460.

[DCLK03] DONG F., CLAPWORTHY G. J., LIN H., KROKOS M. A.: Nonphotorealistic rendering of medical volume data. *IEEE Comput. Graph. Appl. 23*, 4 (2003), 44–52.

[Dec96] DECAUDIN P.: *Cartoon-looking rendering of 3D-scenes*. Technical report 2916, INRIA, June 1996.

[DFR04] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S.: Interactive rendering of suggestive contours with temporal coherence. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2004), ACM Press, pp. 15–145.

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. *ACM Trans. Graph. 22*, 3 (2003), 848–855.

[DS02] DECARLO D., SANTELLA A.: Stylization and abstraction of photographs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 769–776.

[DWE03] DIEPSTRATEN J., WEISKOPF D., ERTL T.: Interactive cutaway illustrations. *Comput. Graph. Forum 22*, 3 (2003), 523–532.

[Elb98] ELBER G.: Line art illustrations of parametric and implicit forms. *IEEE Trans. Vis. Comput. Graph 4*, 1 (1998), 71–81.

[ER00] EBERT D., RHEINGANS P.: Volume illustration: non-photorealistic rendering of volume models. In *VIS '00: Proceedings of the conference on Visualization '00* (Los Alamitos, CA, USA, 2000), IEEE Computer Society Press, pp. 195–202.

[FAFH90] FOLEY J. D., A. VAN DAM, FEINER S. K., HUGHES J. F.: *Computer Graphics—Principles and Practice*, second ed. Addition Wesley, 1990.

[FMS02] FREUDENBERG B., MASUCH M., STROTHOTTE T.: Real-time halftoning: a primitive for non-photorealistic shading. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2002), Eurographics Association, pp. 227–232.

[GCS02] GOOCH B., COOMBE G., SHIRLEY P.: Artistic vision: painterly rendering using computer vision techniques. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2002), ACM Press, pp. 83–ff.

[GGSC98] GOOCH A., GOOCH B., SHIRLEY P., COHEN E.: A non-photorealistic lighting model for automatic technical illustration. *Computer Graphics 32*, Annual Conference Series (1998), 447–452.

[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 *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2000), ACM Press, pp. 43–52.

[GRG04] GOOCH B., REINHARD E., GOOCH A.: Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph. 23*, 1 (2004), 27–44.

[GSG*99] GOOCH B., SLOAN P.-P. J., GOOCH A., SHIRLEY P., RIESENFELD R.: Interactive technical illustration. In *SI3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics* (New York, NY, USA, 1999), ACM Press, pp. 31–38.

[HBH03] HADWIGER M., BERGER C., HAUSER H.: High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 40.

[HL94] HSU S. C., LEE I. H. H.: Drawing and animation using skeletal strokes. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM Press, pp. 109–118.

[HLW93] HSU S. C., LEE I. H. H., WISEMAN N. E.: Skeletal strokes. In *UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology* (New York, NY, USA, 1993), ACM Press, pp. 197–206.

[HZ00] HERTZMANN A., ZORIN D.: Illustrating smooth surfaces. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 517–526.

[iAH03] ICHI ANJYO K., HIRAMITSU K.: Stylized highlights for cartoon rendering and animation. *IEEE Computer Graphics and Applications 23*, 4 (July/Aug. 2003), 54–61.

[Int97] INTERRANTE V.: Illustrating surface shape in volume data via principal direction-driven 3d line integral convolution. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 109–116.

[KDMF03] KALNINS R. D., DAVIDSON P. L., MARKOSIAN L., FINKELSTEIN A.: Coherent stylized silhouettes. *ACM Trans. Graph. 22*, 3 (2003), 856–861.

[KGC00] KAPLAN M., GOOCH B., COHEN E.: Interactive artistic rendering. In *NPAR* (2000), pp. 67–74.

[KLK*00] KLEIN A. W., LI W. W., KAZHDAN M. M., CORREA W. T., FINKELSTEIN A., FUNKHOUSER T. A.: Non-photorealistic virtual environments. In *Proceedings of ACM SIGGRAPH 2000* (July 2000), pp. 527–534.

[KWTM03] KINDLMANN G., WHITAKER R., TASDIZEN T., MOLLER T.: Curvature-based transfer functions for direct volume rendering: Methods and applications. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)* (Washington, DC, USA, 2003), IEEE Computer Society, p. 67.

[Lei94] LEISTER W.: Computer generated copper plates. *Computer Graphics Forum 13* (1994), 69–77.

[LM02] LUM E. B., MA K.-L.: Hardware-accelerated parallel non-photorealistic volume rendering. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2002), ACM Press, pp. 67–ff.

[LME*02] LU A., MORRIS C. J., EBERT D., RHEINGANS P., HANSEN C.: Non-photorealistic volume rendering using stippling techniques. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society.

[LMHB00] LAKE A., MARSHALL C., HARRIS M., BLACKSTEIN M.: Stylized rendering techniques for scalable real-time 3d animation. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2000), ACM Press, pp. 13–20.

[LPFH01] LENGYEL J., PRAUN E., FINKELSTEIN A., HOPPE H.: Real-time fur over arbitrary surfaces. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics* (New York, NY, USA, 2001), ACM Press, pp. 227–232.

[LS95] LANSDOWN J., SCHOFIELD S.: Expressive rendering: A review of nonphotorealistic techniques. *IEEE Comput. Graph. Appl. 15*, 3 (1995), 29–37.

[MBC02] MITCHELL J. L., BRENNAN C., CARD D.: Real-time image-space outlining for non-photorealistic rendering. In *SIGGRAPH* (2002).

[MH04] MCGUIRE M., HUGHES J. F.: Hardware-determined feature edges. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2004), ACM Press, pp. 35–147.

[MKT*97] MARKOSIAN L., KOWALSKI M. A., TRYCHIN S. J., BOURDEV L. D., GOLDSTEIN D., HUGHES J. F.: Real-time nonphotorealistic rendering. *Computer Graphics 31*, Annual Conference Series (1997), 415–420.

[ND03] NIENHAUS M., DÖLLNER J.: Edge-enhancement - an algorithm for real-time non-photorealistic rendering. In *WSCG* (2003).

[NK04] NAGY Z., KLEIN R.: High-quality silhouette illustration for texture-based volume rendering. In *Journal of WSCG* (Plzen, Czech Republic, Feb. 2004), Skala V., (Ed.), vol. 12, UNION Agency - Science Press.

[NM00] NORTHRUP J. D., MARKOSIAN L.: Artistic silhouettes: a hybrid approach. In *NPAR '00: Proceedings of the 1st international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2000), ACM Press, pp. 31–37.

[NSW02] NAGY Z., SCHNEIDER J., WESTERMANN R.: Interactive volume illustration. In *VMV* (2002), pp. 497–504.

[Ost99] OSTROMOUKHOV V.: Digital facial engraving. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), ACM Press/Addison-Wesley Publishing Co., pp. 417–424.

[PDB*01] POP M., DUNCAN C., BAREQUET G., GOODRICH M., HUANG W., KUMAR S.: Efficient perspective-accurate silhouette computation and applications. In *SCG '01: Proceedings of the seventeenth annual symposium on Computational geometry* (New York, NY, USA, 2001), ACM Press, pp. 60–68.

[PHWF01] PRAUN E., HOPPE H., WEBB M., FINKELSTEIN A.: Real-time hatching. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM Press, p. 581.

[Ras01] RASKAR R.: Hardware support for non-photorealistic rendering. In *HWWS '01: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware* (New York, NY, USA, 2001), ACM Press, pp. 41–47.

[RC99] RASKAR R., COHEN M.: Image precision silhouette edges. In *SI3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics* (New York, NY, USA, 1999), ACM Press, pp. 135–140.

[RK00] R´OSSL C., KOBBELT L.: Line-art rendering of 3d-models. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2000), IEEE Computer Society, p. 87.

[RTF*04] RASKAR R., TAN K.-H., FERIS R., YU J., TURK M.: Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph. 23*, 3 (2004), 679–688.

[RvE92] ROSSIGNAC J., VAN EMMERIK M.: Hidden contours on a frame-buffer. In *Proceedings of the 7th Eurographics Workshop on Computer Graphics Hardware* (Cambridge, UK, September 1992), pp. 188–204.

[SABS94] SALISBURY M. P., ANDERSON S. E., BARZEL R., SALESIN D. H.: Interactive pen-and-ink illustration. In *SIGGRAPH '94: Proceedings of the 21st annual* *conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM Press, pp. 101–108.

[SD04] SANTELLA A., DECARLO D.: Visual interest and npr: an evaluation and manifesto. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2004), ACM Press, pp. 71–150.

[SE03] SVAKHINE N. A., EBERT D. S.: Interactive volume illustration and feature halos. In *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2003), IEEE Computer Society, p. 347.

[SFWS03] SOUSA M. C., FOSTER K., WYVILL B., SAMAVATI F. F.: Precise ink drawing of 3D models. *Comput. Graph. Forum 22*, 3 (2003), 369–380.

[SGG*00] SANDER P. V., GU X., GORTLER S. J., HOPPE H., SNYDER J.: Silhouette clipping. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2000), ACM Press/Addison-Wesley Publishing Co., pp. 327–334.

[SLM02] STOMPEL A., LUM E. B., MA K.-L.: Visualization of multidimensional, multivariate volume data using hardware-accelerated non-photorealistic rendering techniques. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2002), IEEE Computer Society, p. 394.

[SSLR96] SCHUMANN J., STROTHOTTE T., LASER S., RAAB A.: Assessing the effect of non-photorealistic rendered images in cad. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 1996), ACM Press, pp. 35–41.

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1990), ACM Press, pp. 197–206.

[SWHS97] SALISBURY M. P., WONG M. T., HUGHES J. F., SALESIN D. H.: Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 401–406.

[TC00] TREAVETT S. M. F., CHEN M.: Pen-and-ink rendering in volume visualisation. In *VIS '00: Proceedings of the conference on Visualization '00* (Los Alamitos, CA, USA, 2000), IEEE Computer Society Press, pp. 203–210.

[VKG04] VIOLA I., KANITSAR A., GROLLER M. E.: Importance-driven volume rendering. In *VIS '04: Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 139–146.

[WPFH02] WEBB M., PRAUN E., FINKELSTEIN A., HOPPE H.: Fine tone control in hardware hatching. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2002), ACM Press, pp. 53–ff.

[WS94] WINKENBACH G., SALESIN D. H.: Computer-generated pen-and-ink illustration. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), ACM Press, pp. 91–100.

[WS96] WINKENBACH G., SALESIN D. H.: Rendering parametric surfaces in pen and ink. In *Proceedings of SIGGRAPH 96* (Aug. 1996), Computer Graphics Proceedings, Annual Conference Series, pp. 469–476.

[XC04] XU H., CHEN B.: Stylized rendering of 3d scanned real world environments. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (New York, NY, USA, 2004), ACM Press, pp. 25–34.

[ZISS04] ZANDER J., ISENBERG T., SCHLECHTWEG S., STROTHOTTE T.: High quality hatching. *Computer Graphics Forum 23*, 3 (Sept. 2004), 421–430.

[ZKEH97] ZHANG H., KENNETH E. HOFF I.: Fast backface culling using normal masks. In *SI3D '97: Proceedings of the 1997 symposium on Interactive 3D graphics* (New York, NY, USA, 1997), ACM Press, pp. 103–ff.