Pacific Graphics 2008
T.Igarashi,N. Max, and F. Sillion
(Guest Editors)

Volume 27(2008), Number 7

Online Personalised Non-photorealistic Rendering Technique for 3D Geometry from Incremental Sketching

Daychyi Ku[1], Shengfeng Qin[†1], David K. Wright[1] and Cuixia Ma[2]

[1]School of Engineering and Design, Brunel University, Uxbridge, UB8 3PH, UK
[2]Institute of Software, Chinese Academy of Sciences, Beijing, 100080, China

ABSTRACT

This paper presents an online personalised non-photorealistic rendering (NPR) technique for 3D models generated from interactively sketched input. This technique has been integrated into a sketch-based modelling system. It lets users interact with computers by drawing naturally, without specifying the number, order, or direction of strokes. After sketches are interpreted as 3D objects, they can be rendered with personalised drawing styles so that the reconstructed 3D model can be presented in a sketchy style similar in appearance to what have been drawn for the 3D model. This technique captures the user's drawing style without using template or prior knowledge of the sketching style. The personalised rendering style can be applied to both visible and initially invisible geometry. The rendering strokes are intelligently selected from the input sketches and mapped to edges of the 3D object. In addition, non-geometric information such as surface textures can be added to the recognised object in different sketching modes. This will integrate sketch-based incremental 3D modelling and NPR into conceptual design.

Keywords: Non-photorealistic rendering, incremental sketching, 3D geometry modelling, conceptual design.

ACM CCS: H.5.2 [Information Interface and Presentation] Interaction Styles, J.6 [Computer-Aided Engineering] Computer-aided design (CAD).

1. Introduction

Sketching is inevitably an important support for the early design stage not only because sketching with a pen is still more natural than a CAD system with keyboard or mouse, but also that sketches are informal figures often created as a way of thinking about, or working through a problem [Dav07]. The most striking feature of hand-drawn sketches is their "incorrectness" such as wiggliness, overtracing and overshooting. People are more likely to discuss design variations with sketches. Schumann and others [SSRL96] show that hand-drawn images or sketches are more appropriate for use in the early design phase and for discussion with customers. Users prefer freehand sketch style drawings for conceptual design rather than precise line drawings in the traditional CAD approach such as 2D drafting.

Goel [Goe95] shows that the freehand sketches that are produced in the conceptual design phase are necessarily and generally vague and ambiguous. Lim [LLD01] argues that conceptual sketches include a variety of vague Information and further suggests that the vagueness in sketches needs to be preserved as long as possible during the early design process. The vagueness can exist in terms of texts, symbols as well as sketches.

Most traditional CAD systems are not suitable in the conceptual stage since they require complete, concrete and precise design information input, which is only available at the end of the design process. Therefore, for conceptual design there is a need to develop appropriate techniques for modelling and rendering 3D objects from freehand sketches. They can allow users to interact with computers by freehand drawing, offering a freedom not available with traditional CAD systems. In addition, computerised information (processed models and drawings) can be visualised in a sketchy appearance similar to input sketches and carried forward in an incremental design process.

One approach that allows the maintenance of the appearance of the processed model in the original sketch style drawn by designers is that of non-photorealistic rendering (NPR). In particular, studies such as [KMK*02][MSK02] have proposed and argued for the use of NPR on recon-

---

[†] Corresponding author

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.
Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

structed 3D models as a means of minimising distraction and leaving "the designer's creative process undisturbed."

This paper presents an online personalised non-photorealistic rendering (NPR) technique for 3D models generated from sketched input. This technique has been integrated into a sketch-based modelling system. The personalised NPR (PNPR) technique lets users interact with computers by drawing naturally and accept overtracing and overshooting strokes. In comparison with typical NPR techniques developed in past studies where 3D models were rendered based on templates or palettes that predefine stroke styles, the main contributions of this paper are

- A new way of integrating sketch based modelling and personalised sketch rendering into an incremental design process. It allows designers to be able to draw as they would normally, without specifying the number, order, or direction of strokes in a drawing. It also allows designers to sketch out both geometric and non-geometric information for rapid conceptual design communications.
- Our PNPR technique doesn't need any templates or predefined stroke styles. It enables the user to be able to sketch a model, have it reconstructed and be able to view it from different points of view while maintaining the user's sketching style. The drawing style is learnt from an online sketch session and thus personalised. Furthermore, it can be applied to both the frontal and hidden geometry.

This paper is structured as follows. After a literature review, a brief introduction of the sketch-based system is presented. The paper then focuses on modelling personalised drawing styles and their applications. Following this, a personalised rendering model is described with examples. Finally, conclusions are drawn.

2. Related work

In general, research in NPR can be categorised into two groups: *2D image rendering* and *rendering of 3D scenes*. Our research work is concerned with 3D rendering. The research relating to 3D scene rendering concerns the appearance of 3D geometry rendered in artistic forms. Current research in this field has investigated the stroke-based rendering approach where *sketchy* strokes are represented by 2D vectors on relevant 3D surfaces or silhouettes.

Research in 3D NPR can be categorised into two groups: styles rendering on *ready-made* 3D objects/scenes that are generated by external software [Hal99] [HS99] [NM00] [SP03] and reconstructing and rendering 3D objects/scenes from *freehand sketches*. This review is focused on the latter. Harold [CHZ00] used three predefined stroke styles (skeletal strokes) to draw strokes on billboards, grounds or terrain to provide a world with 3D 'drawings' while maintaining a hand-drawn appearance during navigation. However, Harold's technique is not suitable for representing certain classes of 3D objects, especially geometrically regular or extremely asymmetric objects. In [BCD01], via its curvature estimation scheme 2D silhouette strokes are interpreted as local surfaces. This mechanism permits efficient stroke-based rendering of the silhouette from different viewpoints. However, this system is similar to that of Harold, although good for illustration and annotation in 3D, but less suitable for sketch-based conceptual design work because their modelling ability is very limited. In [ZS03], the goal of accepting sketch input and producing matching sketchy outputs is achieved. The user first sketches out a 3D skeleton with 'blobs'; then switches the system to the refining mode to draw three types of lines (silhouette, line on the skeleton surface and line growing out of the surface) for rendering with sketch. However, this approach cannot be used to directly produce sketchy rendering.

Other studies have investigated the use of NPR in reconstruction algorithms through gestural commands. For example, SKETCH [ZHH96] focuses on 3D rectilinear reconstruction and rendering in NPR. Teddy [IMT99] takes in 2D closed contours that are reconstructed into 3D freeform model. Their NPR approaches are based on predefined templates where reconstructed 3D models can be rendered in the same sketchy style regardless of how the initial freehand sketches are drawn.

3D Sketch [MSK02] renders input strokes in a similar way to the approach proposed in this paper. In particular, the approach in [MSK02] preserves a sketching style in the 3D object by mapping the sketched strokes to a mesh model. However, it is limited to the rendering of simple rectilinear objects and is only able to render the frontal geometry. The approach therefore lacks generality in rendering the hidden parts of reconstructed 3D models. Other NPR approaches such as in [KMK*02][SC04] render strokes by using a predefined "brush" style or by means of templates that can be selected, i.e., reconstructed 3D models will be rendered in a similar sketchy style regardless of how edges in the initial freehand sketches representing 2D projections of 3D models are drawn. Furthermore, the approach in [KMK*02] is limited to rendering readymade 3D objects only, i.e., it is more like a 3D animator tool.

3. Sketch-based modelling and rendering process

The key features of our approach are the consistent integration of sketch-based modelling and personalised rendering and the ability to directly capture personalised drawing styles from modelling sketches. Meanwhile, our approach aims to support sketchy rendering for both geometric and non-geometric information. As pointed in [Dav07], the difficulty of sketch interpretation increases with users' degree of freedom. The more constrained the draw style, the less difficult the interpretation task. Also, sketch interpretation is context-sensitive, requiring an understanding of the domain knowledge and application contexts.

For our work, on the one hand, we want people to be able to draw without specifying the number, order, or direction of strokes in a drawing. On the other hand, we want to allow a 3D model (design) to be described in different ways such as special symbols (Figure 1a), dimensions and

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

other annotations (Figure 1b). In order to get a balance, we set two modes for sketching: the profile mode and the hatching mode.

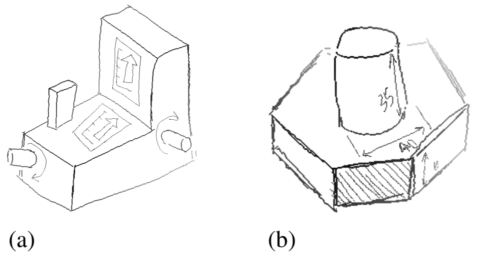


Figure 1: *Sketch contexts*

The overall sketch-based modelling and rendering process is shown in Figure 2. The user starts drawing in the profile mode and then begins to build up 3D geometry incrementally with the addition of PNPR rendering. During this modelling process, the user can switch the drawing mode to hatching and make changes to appearance as shown in Figure 1b. Then, the user can return to the profile mode to continue the design modelling process. This process is iterative.

For the rendering of profile lines, the PNPR algorithm involves the mapping of sketched strokes to their associated edges of the object obtained from the *profile line* sketching mode. This mapping process for the frontal geometry is done directly from the original sketch strokes. However, the silhouette lines for a surface from certain viewpoints must be mapped according to the visibility of the edges, i.e., the edges could be partially or entirely blocked by visible faces. While the object is rotated, the visible portion of strokes belonging to a visible edge changes. Strokes mapped onto a visible edge need to be repositioned after each transformation.

Note that the rendering process needs to take into account strokes generated for rendering hidden geometry as those strokes are not available in the initial sketch, since the input sketch is a natural line drawing without hidden lines, as shown in Figure 2a. The hidden geometry is obtained from the 3D reconstruction process based on some assumptions. Therefore, our PNPR algorithm needs to generate rendering strokes from the initial sketches that can render both frontal geometry edges and hidden edges in a similar appearance to what the front edges initially are drawn. The object will be rendered in a sketch style according to the original input sketch. However, the rendering style changes with the sketching style used for each object. The main advantage of PNPR is that it does not require predefined templates or prior-knowledge of the sketching style, but instead, models the sketching style according to how the input sketch is drawn.

In the PNPR process, strokes drawn in the *hatching* mode are interpreted as details on 3D surfaces. The hatching lines are mapped directly onto the 3D surfaces (Figure 2h), and move with the surfaces during transformation. The visibility of the surface determines the visibility of the hatching lines mapped to that surface.

Note that the mapping strokes for straight-line edges are handled separately from curve edges in the PNPR process. The PNPR process is carried out in two stages: frontal geometry strokes mapping and hidden geometry strokes mapping. Details of the PNPR process are discussed in the following sections.

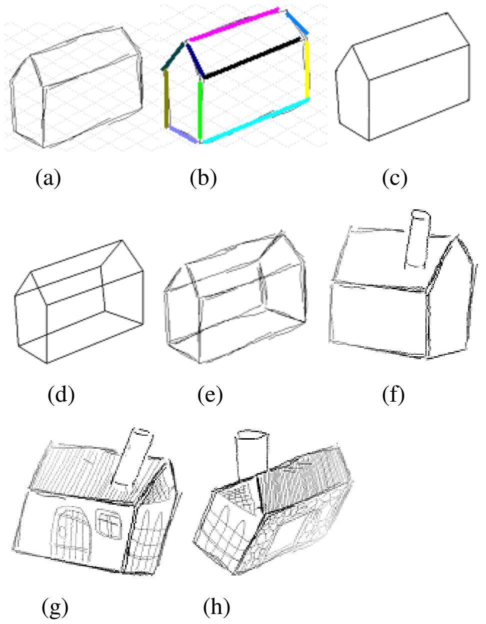


Figure 2: *Sketch-based modelling and PNPR process: (a) freehand sketching, (b) grouping 2D sketches into different segments, and fitting each group of strokes into an conic curve (colour lines), (c) These fitted segments tidied-up in 2D with proper connections and other constraints, (d) 3D reconstruction of both frontal and hidden geometry, (e) capturing PNPR styles and applying them back to processed geometry, (f) after a view change, the geometry rendered in its sketchy form and ready for incremental modelling input (a cylinder), (g) switching to the hatching model and drawing some strokes on the surfaces, (h), after a view change, the geometry rendered with surface details as well.*

In the geometric modelling process, the user can sketch with discontinuous, overlapping and overtracing strokes. Each stroke is a sketch generated between a mouse button down and its release. The input sketch is a natural line drawing with hidden lines removed that depicts a 3D object in an isometric view. The line drawing is interpreted by a series of stroke grouping and 2D tidy-up processes to produce a vertex-edge graph for 3D reconstruction (more details are discussed in [KQW06a]). A novel reconstruction approach based on three-line-junction analysis and planarity constraints is then used to approximate the 3D geometry. After that, the personalised drawing styles can be captured and re-used to render the 3D object when viewed

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

from different view points (Figure 2e-f). A new sketch can then be added to the existing 3D object, and reconstructed into 3D by referring to the existing 3D object from the current viewpoint. This incremental modelling enables a 3D object to be reconstructed from multiple sketching sessions from different viewpoints. However, the interface is limited to reconstructing trihedrons from sketches without T-junctions to avoid ambiguity in the hidden topology determination. In addition, feature-based object recognition for simple curved objects such as cylinders and cones has been integrated into this modelling process. The reconstruction algorithm is detailed in [KQW06b].

In the hatching mode, the user can sketch out symbols, texts, surface textures and other engineering and design information. Sketches drawn on the surface of the object will be mapped on the surface (Figure 2g). The sketches drawn out of the surface will be trimmed off after the model is rotated. The 3D object can then be transformed so that it can be viewed from different viewpoints with an appearance similar to that of the original sketches.

4. PNPR method

4.1 PNPR style definition

Before a 3D object can be rendered with a NPR style, the style components need to be identified and the corresponding style needs to be defined. PNPR aims to model the drawing style from the freehand sketch. In particular, it is observed that the most distinctive feature in a freehand sketch is its "incorrectness". For example, lines drawn by freehand are never perfectly straight (unless a tool such as a ruler is used). Instead, these lines tend to be wiggly and overshoot at the intended endpoints.

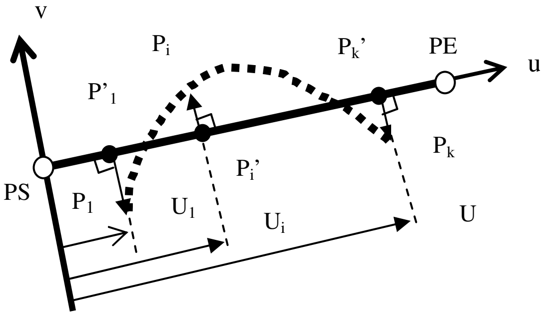


Figure 3: *Stroke normalisation based on fitted parametric line segment.*

Here, the focus of the PNPR algorithm is on modelling the *distortion* of a sketched line, i.e., the degree to which the line deviates from the intended *analytical path.* This is observed as the degree of wiggliness in the sketched line. In addition, other imperfections that are modelled include the position of a sketch stroke relative to the analytical path, the relative line length, and the variation in ending points compared to the intended ending points. These distortions and imperfections compared to the analytical path are described collectively as the *rendering style*, which is used to model different sketching styles according to how sketches are drawn. The modelling process of how freehand sketches are drawn is essentially a parameterisation process of sketched strokes based on the analytical path.

To model the *sketchy style*, the distortions of strokes are parameterised with equations. The PNPR style model for an edge is defined by three components:

1. An analytical path comes from the 2D-tidy-up process, which is the intended edge geometry from a group of strokes. It is used as a reference to parameterise associated hand-drawn strokes.
2. Parameterised strokes. Each stroke is processed by referring to its analytical path. The distortions of the stroke are used to model the sketching style that can be different from one individual to another. Figure 3 illustrates how these distortions are captured. Let the stroke consist of k points $P_i | i = 1, 2, 3, \ldots, k$ and their projection points on the analytical path be $P'_i | i = 1, 2, 3, \ldots, k$. PS and PE are the starting and ending points of the path, respectively. A local u-v coordinate system can be established with the origin at PS, the u-axis pointing to PE, and the v axis being perpendicular to the u-axis. The data points can be represented by $U_i | i = 1, 2, 3, \ldots, k$, reflecting the relative position of each point on the analytical path and $V_i | i = 1, 2, 3, \ldots, k$, describing the level of deviation of the stroke. U_i and V_i are the parameters of the associated stroke on the u and v-axis respectively.

   The parameters U_i and V_i can be further normalised to obtain their relative positions to the analytical path as E_i and F_i respectively.
3. The number of associated strokes. For a visible edge this is obtained from the 2D tidy-up process.

4.2 Straight-line rendering

4.2.1 *Frontal geometry*

The straight-lines of 3D frontal geometry are rendered by mapping the original strokes onto the associated edges in the current viewpoint. This is a un-normalisation process for associated strokes against a new edge.

During the process, the change in the position and length of the analytical line segment is used to calculate the new position of P_i from E_i and F_i. Let the ratio of the new length to the original length be R. The points on new analytical line segment with new endpoints PS_n and PE_n are

$$P'_{i(map)} = PS_n + E_i(PE_n - PS_n) \qquad (1)$$

The sketched stroke being mapped to the new analytical line segment becomes

$$P_{i(map)} = P'_{i(map)} + RF_i \bullet (\overrightarrow{z} \times \frac{(PE_n - PS_n)}{|PE_n - PS_n|}) \qquad (2)$$

Where z is the unit vector normal to the screen.

The position and length for a visible edge will be changed during transformation of the 3D object. As such, the mapping of the stroke must be updated after each transformation. New positions of the mapping stroke are calculated by substituting new values of R, PE_n and PS_n into equations 1 and 2. The number of strokes per edge is determined from the grouping process during the 2D tidy-up process.

4.2.2 Hidden Geometry

The hidden geometry of an object is approximated based on some assumptions in the 3D reconstruction process [KQW06b]. The hidden geometry may become visible after transformation. The rendering style for strokes mapped onto the hidden edges needs to be maintained as with the frontal geometry to keep the appearance of the scene consistent. To do so, the appropriate strokes from the original sketch are mapped to the hidden part after having been properly scaled and transformed. The process of generating the PNPR strokes for a hidden edge includes the following steps:

(1) Generate a pool of all sketch strokes.

(2) Produce a candidate stroke pool based on the parallelism check between the hidden edge and all visible edges.

(3) Compute the required number of rendering strokes and if it is larger than the total number of candidate strokes, add all strokes on the same face with the hidden edge to the candidate pool.

(4) Randomly select the required number of candidate strokes according to their weights.

(5) Map (transform) each candidate stroke to render the hidden edge.

The visible line segments having the same slope value as a hidden line segment are first identified. In practice, lines with an identical slope value rarely occur because the input is from a freehand sketch. As such, a tolerance value (threshold) is used to determine the lines with similar slope values. The lines with similar slope values within the tolerance will be grouped as *candidate lines* and their associated sketch strokes are placed in the candidate stroke pool. There is normally more than one candidate line being grouped. Each candidate line is assigned a weight value according to its geometry relationship to the hidden edge. The lines located on the same face with the hidden edge are given higher weight values compared to the lines on the other faces. The weight values will be used to determine the number of strokes on the hidden edge and the selection of candidate strokes.

Let the candidate lines be $L_i|i = 1, 2, 3, \ldots, n$, with n being the total number of candidate lines. The weight values for the candidate lines are $W_i|i = 1, 2, 3, \ldots, n$ and the number of sketched strokes on the line L_i be $N_i|i = 1, 2, 3, \ldots, n$. $W_i = \theta_i / 90 + \Delta$, where θ_i is the smaller angle between the line L_i and the hidden edge, ranging from 0 to 90 degree, Δ is a minimum weight threshold. The number of strokes on the hidden edge is

$$N_{hidden} = \sum_{i=1}^{n} N_i W_i / \sum_{i=1}^{n} W_i$$

The strokes on the candidate lines will inherit the weight values and form a pool of *candidate strokes*. The pool is a subset of the *sketched stroke pool* and is customised for the particular hidden edge. The sketched stroke pool consists of all sketched strokes.

For the case of N_{hidden} less than the total number of strokes in the *sketched stroke pool*, not all the candidate strokes will be used for the hidden edge rendering. The rendering strokes need to be selected from the candidate strokes. A random factor is used to introduce noise into the selection of new rendering strokes, which would enable different strokes to be selected for rendering when the selection is repeated. The random selection involves a random number generator and the pool of weighted candidate strokes. The probability of a rendering stroke being selected is dependent on its weight value. Strokes with higher weight values are more likely to be selected, i.e., the candidate strokes on the same face with the hidden edge are more likely to be selected as the rendering strokes. The weighted, random selection of candidate strokes may allow for a more realistic hidden edge rendering for the output by retaining the characteristic of the rendering style of frontal geometry edges with some small degree of variation.

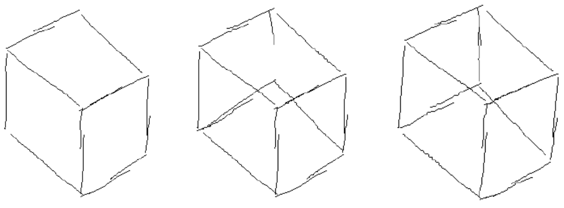

(a) (b) (c)

Figure 5: *(a) Initial sketch, (b) incomplete hidden line rendering, (c) corrected result.*

After that, the hidden edge is checked to determine if it is completely rendered with sketch strokes, i.e., collective mapping of strokes should cover more than 80% (a threshold) of the total length of the segment. If there is any uncovered gap of more than 20% (a threshold) of the total length, extra candidate strokes with an appropriate position (the stroke containing U_i value where the gap appears) are mapped to the hidden edge, as shown in Figure 5. Figure 5b shows the incomplete rendering of the hidden geometry. The corrected result is shown in Figure 5c.

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

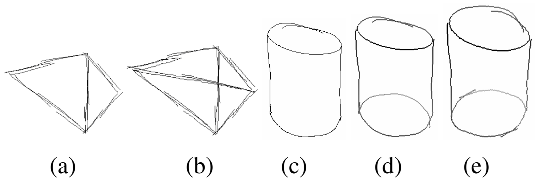


Figure 6: *(a) Initial sketch of a tetrahedron, (b) PNPR hidden line rendering (c) intial sketch of a cylinder, (d) approximated 3D sketched stroke for the top and bottom faces stroke mapping, (e) rotated cylinder.*

If there is no suitable stroke to fill in the gaps, this implies that such a gap exists in the frontal geometry of the original sketch. This could be a specific sketching style from the user or is done purposely with specific reasons. In this case, the gap will be preserved in the hidden edges. The incomplete mapping occurs when the following conditions are met:

1. Some of the candidate strokes cover only part of the line.
2. The number of strokes on the hidden edge, N_{hidden}, is too small for the complete mapping. This is particularly true when the candidate lines on the same face have smaller N_i than the candidate lines on the other faces, while the rendering strokes selected are mainly from the latter lines.
3. When the same candidate stroke is selected for more than once to render the hidden edge. However, this situation does not always occur.

If there is no candidate line that passes the slope test with the hidden edge, all the lines on the same faces to the hidden edge are selected as candidate lines and assigned the same weight values. Figure 6a shows an example of such an object and the corresponding PNPR for the hidden geometry (Figure 6b).

4.3 Revolve Objects Rendering

The revolve objects here referred to arc cylinder and cone. There are two types of rendering strokes for the objects: *silhouette of the cylindrical surface* and *edges of the top and bottom faces* of the objects. The silhouettes of the objects are rendered in the original sketched strokes with the straight-line rendering technique, as discussed above. During a view transformation, the silhouette strokes are repositioned to the surface's profile of the objects. Figures 6(c-d) show examples of the silhouettes rendering where the strokes used to render cylindrical surface profiles are scaled and repositioned according to the new profile lines.

Instead of using stroke mapping as discussed in the straight-line rendering, the top and bottom faces of the revolve objects are rendered by sketched strokes in 3D vertices. The 3D equations of the faces are obtained in the reconstruction process. The 3D vertices of sketched strokes for curved edges are approximated from the 3D faces.

The bottom face which is partially visible from the original viewpoint (Figure 6c) does not have full strokes for the hidden part rendering. Therefore, we need to generate new strokes for rendering the hidden part of the face. The strokes are generated by inverting the sketched stroke on the visible part of the face, so that the invisible part is mirrored to the visible part of the curve (Figure 6d).

4.4 Depth perception enhancement

The PNPR strokes are rendered in various grey tones for depth perception enhancement. The depth value of each data point on visible strokes is calculated and assigned a grey tone accordingly.

An intensity ramp based on grey levels is used for depth cueing. The intensity of a point is adjusted according to the distance from the designer's viewpoint. The intensity varies linearly from 1.0 to 0.0 with black representing the intensity value 1.0 and white representing the intensity value 0.0. Colour blending is used to diminish the intensity of strokes for the edges that are located further away from the designer's viewpoint.

Note that the camera and light source positions are fixed during transformations of objects. As such, the intensity of the object's edges changes while the object is rotated. That is, the intensity of each of the rendering strokes mapped to the object's edges has to be updated accordingly. To ensure that the intensity of the object's edges is rendered within the range from 1.0 (black) to 0.0 (white) during transformation, the depth range of the entire object is updated in real-time during transformation. The depth range is the difference between the highest and the lowest depth values of the 3D object. The grey tones for depth values are adjusted with respect to the depth range. The depth perception enhancement is applied and can be seen from most of the figures in this paper such as Figure 7.

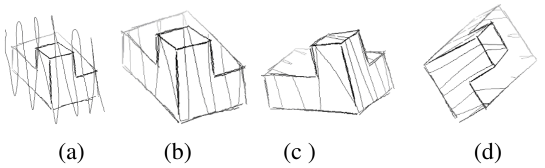
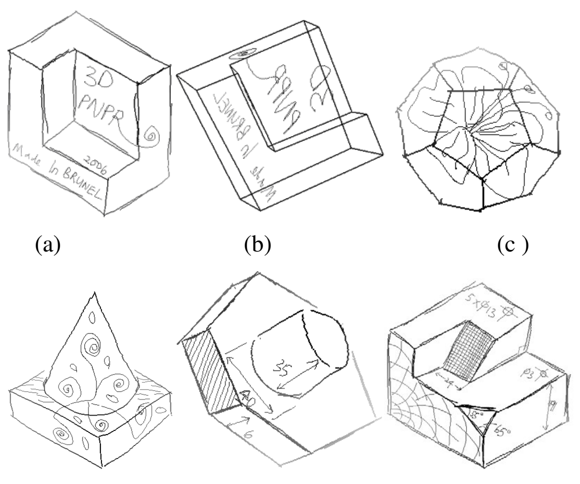


Figure 7: *(a) Stroke sketched in hatching mode, (b) data points of the stroke lying outside a surface are trimmed off, (c) the stroke is segmented into two when it is sketched across a step, (d) viewing from another viewpoint.*

4.5 Non-geometric refinement

Non-geometric information, e.g., hatching (Figure **7**) and annotation (Figure 8), can be added to the reconstructed 3D models to allow more details to be added and presented. In addition, the geometric and non-geometric information can be presented together for communication. This is important as fine-tuning of objects changes the way viewers think about the design itself, e.g., using hatching to provide surface refinements and also to emphasise one particular surface area and distinguish among other areas [DGNZ00][MSK02].

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.



(a) (b) (c) (d) (e) (f)

Figure 8: *(a) PNPR with surface details, (b) parametric wireframe rendering, (c) dodecahedron, (d) model with cone, (e) model with cylinder, (f) polyhedron.*

After 3D objects are reconstructed, details are added by freehand sketching onto the surfaces of the objects through the *hatching* mode. The strokes sketched in this mode will be interpreted as curves lying on 3D surfaces. Strokes sketched outside a surface (entirely or partially) will be trimmed off automatically (Figures 7(a-b)). When a stroke is sketched across a step, i.e., transition in 3D distance that is greater than a threshold, the stroke is segmented into two portions at the location where the step occurs. The segmentation is carried out by dividing the stroke at the proper location to ensure that it is not dangling but instead is placed on the appropriate surface, as shown in Figures 7(c-d).

When a hatching stroke is sketched, the vertices for its data points will be calculated in real-time based on a 3D surface under the isometric projection, in which the angles between the projection of the X, Y, and Z-axes are all the same or 120°. This projection with the Y axis vertical can be obtained by an object rotation about the Y axis by θ=45° followed by a rotation about the X axis with φ=35.25°, which can be represented as the following equation for a point (x', y', z').

$$\begin{bmatrix} i \\ i \\ k \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ \sin\theta\sin\varphi & \cos\varphi & -\cos\theta\sin\varphi \\ -\sin\theta\cos\varphi & \sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

By interpreting the strokes as lying on a surface plane, the vertex (x', y', z') of a data point with a sketch (i, j) on the surface with equation $ax + by + cz = d$ is given by replacing the third row of the above equation with the surface equation and θ=45° and φ=35.25. We then have

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a & b & c \\ 0.7071 & 0 & 0.7071 \\ 0.4082 & 0.8165 & -0.4082 \end{bmatrix}^{-1} \begin{bmatrix} d \\ i \\ j \end{bmatrix}.$$

5. Results and discussion

A prototype system has been developed in Visual C++ 6.0 with MFC in Microsoft Windows XP.

Figure 8 shows results of a variety of reconstructed objects obtained from freehand sketches that are rendered using the PNPR algorithm. The results show that the PNPR algorithm can render reconstructed objects in an appearance similar to the original input freehand sketches. Unlike NPR techniques developed in past studies [KMK*02][MSK02][SC04], the PNPR approach here does not require predefined templates to define specific sketchy stroke or rendering styles for the NPR. Instead, the sketchy stroke style is obtained from the modelling of how the freehand sketch is drawn.

In addition, during conceptual design, it is known that freehand sketches may contain various non-geometry information such as annotation and hatching. The proposed PNPR approach allows for both geometry and non-geometry information in the freehand sketch to be captured through both the profile and hatching sketching modes, respectively, which are then processed to allow for the rendering of the reconstructed 3D model with the associated geometry and non-geometry information in an appearance similar to the original sketch. For example, Figures 8c-d show reconstructed objects with surface refinements drawn in the initial freehand sketches while Figures 8e-f show reconstructed objects with texts and symbols.

6. Conclusion and future work

We have presented a novel PNPR rendering technique for rendering recognised 3D objects from freehand sketches, maintaining the appearance of the objects from multiple viewpoints. To achieve this, we reconstruct the 3D objects from the freehand sketches, followed by mapping the sketched strokes to the appropriate edges. Various grey tones are used to indicate the intensity of light on the edges, thus suggesting the depth cue of the 3D scene. During transformation of objects, the position, orientation and the intensity of the rendering strokes are updated. Hatching and texture strokes can be drawn on the surfaces of 3D objects. The 3D vertices of these strokes are calculated in real-time based on the related face equations.

The PNPR algorithm allows reconstructed 3D models to be rendered in an appearance similar to that of the initial freehand sketches without the use of predefined template or prior knowledge of the sketch style. It is also able to model techniques for drawing freehand sketches to personalise the rendering style in the NPR algorithm. It is different from previous studies that used predefined and fixed templates

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

in the NPR, those limiting the sketch styles that can be rendered. Furthermore, unlike the early work [MSK02], the PNPR approach also allows hidden lines to be rendered based on their relationships to frontal edges in the initial sketches. In addition, it allows for a depth perception enhancement process based on the use of gray tones to represent different levels of light intensity in relation to the distance between the points on edges to the designer to provide a suggestion of depth cue in the 3D scene.

Future work is to integrate different sketch-based modelling techniques [KC06] to reconstruct complex 3D scenes and to use context-based intelligence to automatically switch to an appropriate sketching mode.

References

[BCD01] BOURGUIGNON D., CANI, M., & DRETTAKIS G.: Drawing for illustration and annotation in 3d. *Computer Graphics Forum*, *20*, 3 (2001), 114-123.

[CHZ00] COHEN J. M., HUGHES J. F., ZELEZNIK R. C.: Harold: A world made of drawings. In *Proc. the 1st international symposium on non-photorealistic animation and rendering 2000* (2000), pp.83-90.

[Dav07] DAVIS R.: Magic Paper: Sketch-Understanding Research, *IEEE Computer,* 40, 9(2007), 34-41.

[DGNZ00] DO E. Y., GROSS M. D., NEIMAN B, ZIMRING C.: Intentions in and relations among design drawings, *Design Studies 21,* 5(2000), 483–503.

[Goe95] GOEL V.: *Sketches of thought.* MIT Press, 1995.

[Hal99] HALL P.: Non-photorealistic rrendering by q-mapping. *Computer Graphics Forum*, *18,* 1 (1999), 27-39.

[HS99] HAMEL J., STROTHOTTE T.: Capturing and re-using rendition styles for non-photorealistic rendering. *Computer Graphics Forum*, *18,* 3 (1999), 173-182.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proc. SIGGRAPH 99,* (1999), pp.409-416.

[KC06] KAPLAN M., COHEN E.: Producing models from drawings of curved surfaces, In *3rd eurographics workshop SBIM06,* (2006), pp.51-58.

[KMK*02] KALNINS R. D., MARKOSIAN L., MEIER B. J., KOWALSKI M. A., LEE J. C., DAVIDSON P. L., et al.: Wysiwyg npr: Drawing strokes directly on 3d models. In *SIGGRAPH 02,* (2002), pp. 755-762.

[KQW06a] KU D. C., QIN, S. F., WRIGHT D. K.: Interpretation of overtracing freehand sketching for geometric shapes. In *Proc. WSCG2006,* (2006), pp. 263-270.

[KQW06b] KU D. C., QIN, S. F., WRIGHT D. K. : A sketching interface for 3d modeling of polyhedrons. In *3rd Eurographics workshop SBIM06,* (2006), pp.83-90.

[LLD01] LIM S. W., LEE B. S., DUFFY A. H. B. : Incremental modelling of ambiguous geometric ideas (imagi): Representation and maintenance of vague geometry. *Artificial Intelligence in Engineering 15,* 2 (2001). 93-108.

[MSK02] MITANI J., SUZUKI H., KIMURA F.: 3d sketch: Sketch-based model reconstruction and rendering. *Kluwer International Federation For Information Processing Series*, (2002). 85-98.

[NM00] NORTHUP J. D., MARKOSIAN L.: Artistic silhouettes: a hybrid approach. In *Proc. Npar '00: the 1st international symposium on non-photorealistic animation and rendering,* (2000), pp.31–37.

[SSRL96] SCHUMANN J., STROTHOTTE T., RAAB A., LASER S.: Assessing the effect of non-photorealistic rendered images in cad. In *Proc. chi 96: the 1996 conference on human factors in computing systems,* (1996) pp. 35-41.

[SC04] SHESH A., CHEN B.: Smartpaper: An interactive and user friendly sketching system. *Computer Graphics Forum 23,* 3 (2004). 301-301.

[SP03] SOUSA M. C., PRUSINKIEWICZ P.: A few good lines: Suggestive drawing of 3d models. *Computer Graphics Forum 22,* 3 (2003), 381-390.

[ZHH96] ZELEZNIK R. C., HERNDON K. P., HUGHES J. F.: Sketch: An interface for sketching 3d scenes. In *Proc. SIGGRAPH 96,* (1996), pp. 163-170.

[ZS03] ZENKA R., SLAVIK P.: New dimension for sketches. In *Proc. Sccg '03: Proceedings of the 19th spring conference on computer graphics,* (2003), pp. 157-163.

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.

© 2008 The Author(s)
Journal compilation © 2008 The Eurographics Association and Blackwell Publishing Ltd.