

Technical Section

# A simple normal enhancement technique for interactive non-photorealistic renderings

Paolo Cignoni, Roberto Scopigno, Marco Tarini\*

*Istituto di Scienza e Tecnologia dell'Informazione, Consiglio Nazionale delle Ricerche, Via Moruzzi 1, Pisa 56124, Italy*

---

## Abstract

We present a simple technique to improve the perception of an object's shape. Bump mapping is well known in the computer graphics community for providing the impression of small-scale geometrical features, which are not explicitly modelled. Here, we propose a similar approach (variation of normals) for the purpose of enhancing the perception of a given geometry. Our approach is based on a simple modification of the surface normals in order to enhance the geometric features of the object during the rendering. The enhanced normals produced by this approach can be used in any rendering technique. The technique presented is particularly well suited to improve the rendering of mechanical parts where common straightforward shading techniques can often generate shading ambiguities.

© 2004 Elsevier Ltd. All rights reserved.

*Keywords:* Display algorithms; Non-photorealistic rendering (NPR); Computer-aided design (CAD)

---

## 1. Introduction

In this paper we consider a class of 3D objects, including but not limited to typical mechanical parts used in computer-aided design (CAD) systems, that have a common set of features: flat surfaces, many of which facing the same direction, sharp straight edges, overall regularity. Straightforward rendering of such objects often results in visually unsatisfactory, dull, flat looking, or even unclear and ambiguous images (see Figs. 1 and 2).

Adding enough realism, the problem could disappear: complex realistic effects (common in off line rendering), like cast soft-shadows, inter-reflections, radiosity, local (as opposed to at infinity) light positions, and so on, can produce a much less *flat* result, and are known to provide many intuitive visual hints to the viewer.

In graphic design illustrations (either hand-made, or made with vector-based drawing programs) the problem has been solved in a different, simpler yet effective way: professional illustrators can reduce flatness (or unclarity) “by hand”, shading surfaces according to their esthetic sense (see for example Fig. 1) rather than solving difficult physical problems (shadow projection, light diffused by surfaces, etc.).

The implicit idea behind this is that appropriate shading supplies a kind of information that is more *qualitative* than *quantitative* in the perception of an image. Conversely, the shape of the silhouette and the shading discontinuity bring us the most significant information about the real shape of the object. Moreover, to obtain an improved perception, shading does not have to be physically correct (see Fig. 1).

Along these lines, we designed a new perception-oriented, non-realistic, automatic technique for interactive rendering systems. We aim at synthesizing images that are qualitatively similar to the illustration style visible, for example, in Fig. 2. It is based on

---

\*Corresponding author. Tel.: +39 050 315 2925.

*E-mail addresses:* [cignoni@isti.cnr.it](mailto:cignoni@isti.cnr.it) (P. Cignoni),  
[scopigno@isti.cnr.it](mailto:scopigno@isti.cnr.it) (R. Scopigno), [tarini@iei.pi.cnr.it](mailto:tarini@iei.pi.cnr.it) (M. Tarini).

enhancing high-frequency components of the model; the key issue is that, rather than working on the geometry (vertex positions) of the digital model, we apply the enhancement to the surface orientation alone, leaving the silhouette unchanged. This technique, hereafter called *normal enhancement*, is done on the mesh in a preprocessing stage: the enhanced normals are integrated into the model, making this technique view-independent.

In contrast with most non-photorealistic techniques, this approach is de-coupled from the rendering algorithm used to effectively produce the image. For this reason the enhanced normals can be used into any

rendering subsystem that support user-specified normals, like for example the standard VRML browsers. Moreover, a visualization tool or a geometry browser that uses this technique can easily allow the user to toggle between the normal-enhanced and standard rendering modes.

## 2. Related work

Computer graphics algorithms and techniques that aim to imitate non-photographic illustration styles are usually referred to as *non-photorealistic rendering* [1]

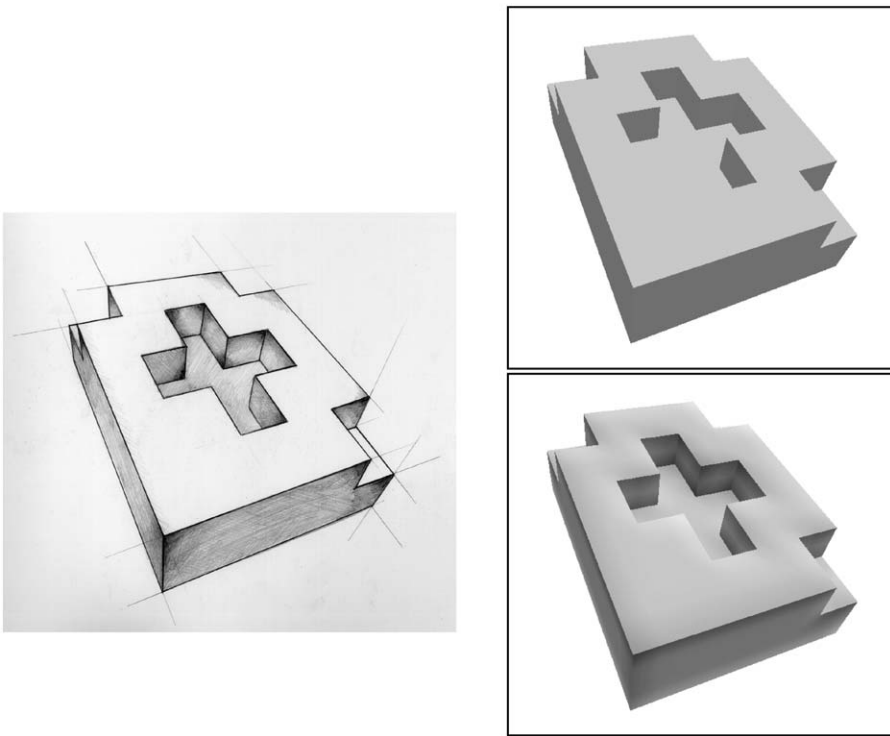


Fig. 1. A hand-drawn pencil drawing with a non-photorealistic shading that enhances the mesh features (left), courtesy of Alessandro Briglia; real-time rendering of a similar object without (top right) and with the proposed method (bottom right).

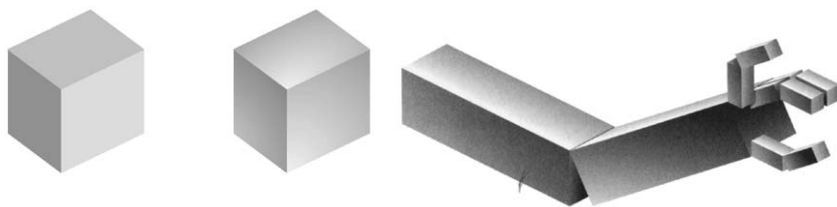


Fig. 2. Examples of non-synthetic, perceptual-oriented drawings of two simple 3D objects: the cube above is drawn in a non-constant, non-realistic manner on the right; the robot arm is a drawing published in Fig. 3.26 of the red book on OpenGL [24].

(NPR). NPR techniques greatly differ in style, visual appearance, and objectives. Many of the presented approaches aim to mimic/imitate some existing artistic techniques or styles like watercolour [2], pen-and-ink [3,4], charcoal [5] or while other works take inspiration from the field of technical and professional illustration. The latter techniques have the main goal of providing a better comprehension of a given 3D structure. Many NPR approaches have been proposed in the last few years making NPR a new branch of computer graphics. A survey on such approaches can be found in [1,6].

Here, for the sake of conciseness, we limit ourselves to review only the papers related more to the field of computer generated rendering of technical illustrations. This approach was probably first explored by Saito and Takahashi [7], who proposed some techniques to enhance the visual comprehensiveness of 3D images by means of some post-rendering image-based processing applied to the final image.

This problem has been faced from a more abstract point of view in some papers [8,9], where without introducing new rendering techniques, the problem of the perception of various kinds of information through the use of computer generated illustrations was discussed.

Gooch et al. [11] presented a non-photorealistic lighting model that provides a better shape comprehension by mapping the change in surface orientation into variations of hue instead of brightness variations [10]. This technique can also be efficiently implemented using current graphics hardware [11].

Another common way, pioneered by the work of [12], consist of detection and outlining of certain elements of the model (like silhouettes and sharp edges). Recently, Raskar [13] proposed a graphic accelerated approach, where some of these elements are drawn in real time without being explicitly detected in a preprocessing phase.

In the general context of the above contributions, the normal enhancement technique presented here can be helpful in generating sharper and less ambiguous images. Moreover, an advantage of this technique is a seamless integration with existing rendering systems.

### 3. Enhancement of mesh features

Using a rather informal signal processing terminology, we can say that to *sharpen* a 3D mesh  $M$  we must enhance the *high-frequency* components of that mesh. A simple way to compute these components is to make a low-pass filtering of  $M$  by means of a Laplacian smoothing kernel [14,15], obtaining a *low-frequency* mesh  $M_L$  (see Fig. 3). Then the *high-frequency* component can be recovered by computing the difference between the original mesh and the smoothed one,

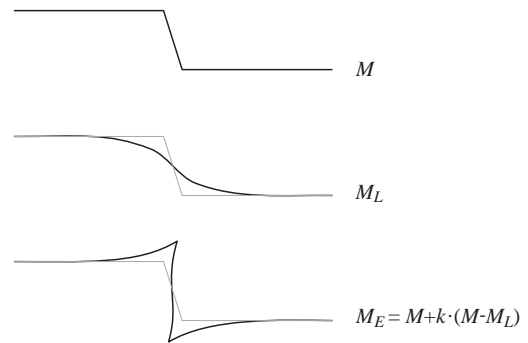


Fig. 3. High-frequency components of a mesh  $M$  can be enhanced by summing to  $M$  the weighted difference between  $M$  and a  $M_L$ , where  $M_L$  is a smoothed representation of  $M$ .

i.e.  $M - M_L$ . The desired result of a *high-frequency* enhancement can be obtained by adding this component to the original mesh, scaled by a user specified constant factor  $k$ : the *enhanced mesh*  $M_E$  is found by  $M_E = M + k(M - M_L)$ .

It must be noted that a more formal and correct signal mesh processing could be done on a generic mesh, as presented for example in a paper by Guskov et al. [16], but our goal is much less elaborate: we want only to enhance the visual presentation of an object in order to improve the perception of some features.

Fig. 4 shows an example of the results obtained by this technique: given a simple input mesh  $M$ , we apply a scale-dependent Laplacian filter [17,14] to obtain a smoothed representation  $M_L$ ; the enhanced mesh  $M_E$  is obtained by summing to each vertex of  $M$  a fraction of the difference vector between that vertex and the corresponding one of  $M_L$ . In  $M_E$  the high-frequency features are visibly enhanced.

However, as we have discussed in Section 1, we do not want to modify the geometry of the input mesh, therefore, we shall describe in the next section how to perform a similar transformation, this time affecting only normal vectors and the shading.

### 4. Normal enhancement

The effects obtained by modifying the geometry of the mesh using the technique discussed in the previous section are interesting and, in some situations, they can be useful. On the other hand, the mesh produced is an object that the users in some sense perceive as inherently different from the original one (at least because the silhouette is changed, see Fig. 4). As noted before, the shading of the surface conveys a lot of *qualitative* information, and we may try to make use of it to improve the perception of surface features.

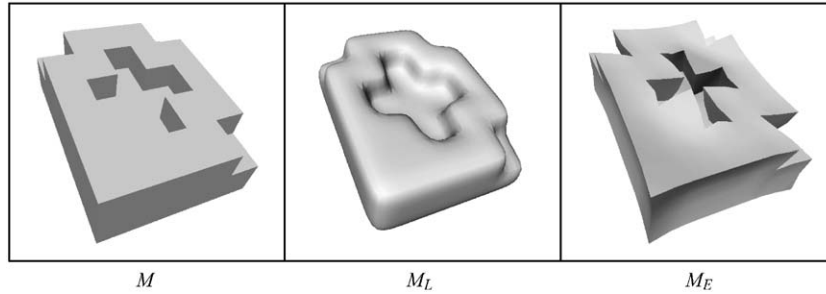


Fig. 4. Enhancement of high-frequency geometry components of a mesh. The object's shape is perceived as inherently different from the original one, and not even the silhouette is preserved.

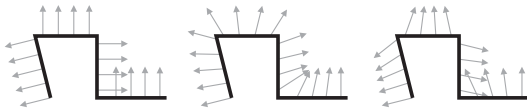


Fig. 5. Enhancement of high-frequency components of a mesh normals vectors.

This is the reason why we propose to modify only the normals of the object instead of the coordinates of the mesh. In this way, affecting only the shaded appearance of the object but keeping its silhouette and its geometry extent, we achieve the desired enhancement in the rendered images without the drawback of having distorted objects in our rendering. The methodology remains analogous to the one described in the previous section, but it is applied only to the surface normals (see Fig. 5) as follows:

- (1) For each face, compute a new normal  $n_L$  as a low-pass filtering of the normals of the mesh; this is done by iteratively substituting each face normal with a re-normalized area-weighted average of the normals of the adjacent faces.
- (2) For each normal vector, enhance it by pushing back the original normal  $n$  in the opposite direction of its averaged counterpart  $n_L$  and then re-normalize the resulting vector, obtaining an enhanced normal  $n_E = n + k \cdot (n - n_L)$ .

Note that the same approach could be obtained also by applying the geometry enhancement technique presented in Section 3 and replacing the normals of the vertices of  $M$  with the normals of the vertices of  $M_E$ . This can be done in a straightforward manner because there is a one-to-one vertex correspondence between  $M$ ,  $M_L$  and  $M_E$ . This technique is somewhat more expensive because it requires storing of both the original and the modified vertex positions. Even if, from an abstract point of view, this approach could be more *correct*, because it allows the exploitation of better smoothing or fairing techni-

ques, we have found that it is more sensitive to the tessellation quality of the starting mesh. We have performed empirical tests and we have not found such an improvement in the final quality of the result to justify the adoption of this latter approach.

The result of the above procedure is a new set of per-face normals. In order to obtain a high-quality shaded rendering it is necessary to correctly compute per-vertex normals. This can be done by using the standard approach of averaging together those face normals of adjacent faces whose normals differ less than a user specified crease angle. In this way the sharp discontinuity of the mesh are preserved while regions with low curvature exhibit a smoother shading.

#### 4.1. Impact of mesh tessellation

The high-frequency enhancement technique (whether applied to the position or to the normal of the vertices) works only if the starting mesh is rather densely and uniformly tessellated or, in other words, if the triangles are small with respect to the size of most of the features of the mesh and the ratio between the largest and the smallest triangle edges is not too large. Also, the connectivity of the mesh must be fairly regular. Common smoothing techniques for triangular meshes do not work well if the mesh lacks the above properties: as intensity distribution across the mesh is not uniform and the displacement of vertices is not controlled, the smoothing process leads to erroneous results.

Therefore, if the starting mesh exhibits a large disparity of triangle sizes, we need to preprocess the mesh by recursively splitting and all the faces larger than a given threshold. In addition, the use of a *scale-dependent* Laplacian filtering [17,14] of normal vectors alleviate the problems arising from uneven tessellation by weighting the normal influence on the neighboring triangles with their size. Another possible solution is to re-mesh the input model, in order to produce a more regular tessellation.

4.2. Bump mapping

When the original object is composed of a small number of faces with many sharp features, our technique needs an initial refinement step. This refinement is needed to ensure the correctness of the

smoothing pass. In some cases, this refinement step can heavily increase the initial complexity of the object (even by a factor of ten or more). In these cases, it may be unacceptable to render, let us say, ten times more triangles to produce an improved rendering. Luckily, the presented technique leaves the surface geometry

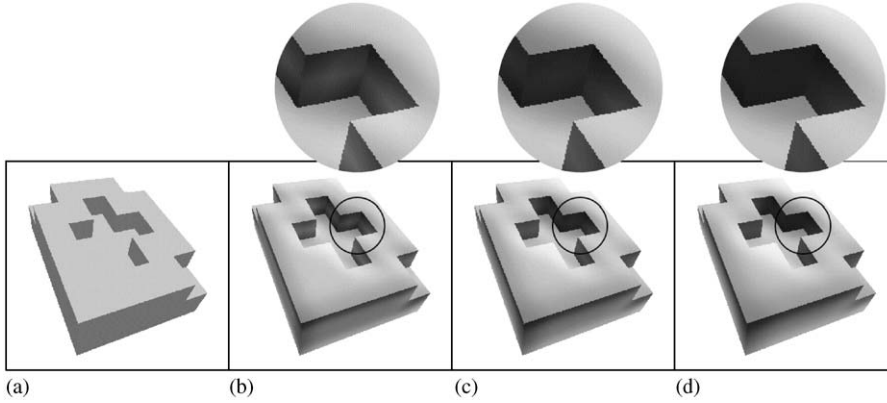


Fig. 6. The amount of normal smoothing affects the normal enhancement of the smallest features: (a) is the original mesh, while (b)–(d) show the effect of normal enhancement by using, respectively, 10, 20 and 30 normal averaging iterations.

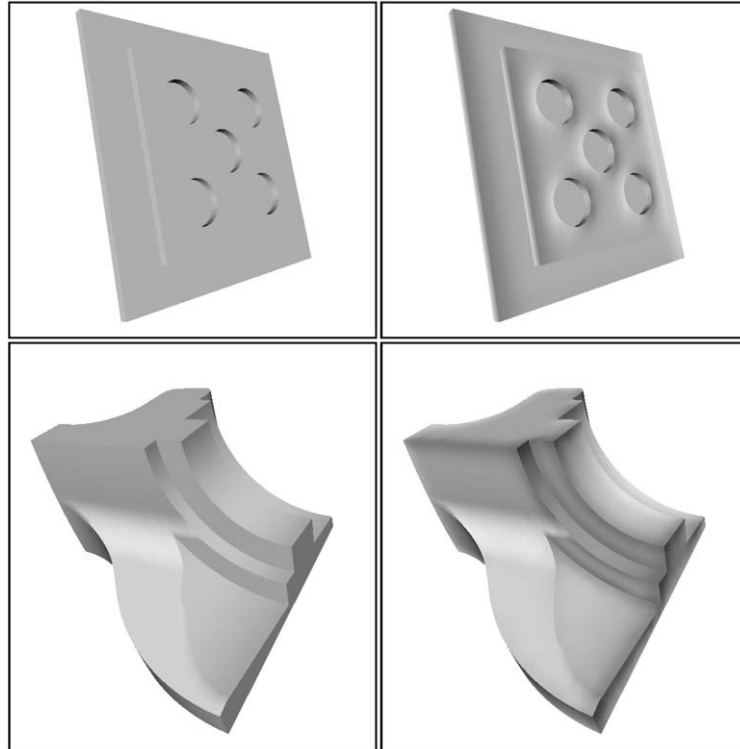


Fig. 7. A side to side comparison between the original mesh (on the left in each subfigure), and the mesh obtained applying enhancement of normals (on the right). Plain, standard renderings of normal enhanced meshes produce clearer images.

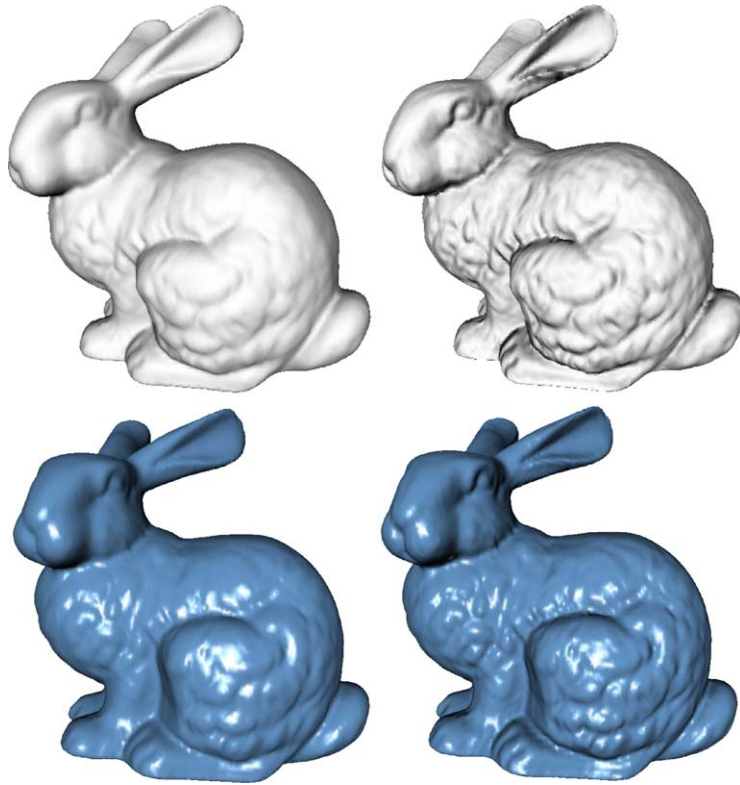


Fig. 8. Enhancement of high-frequency geometry components of an irregularly shaped object (left: original mesh); a diffuse shading is in the top images, while a more specular material is adopted in the ones on the bottom.

untouched and modifies just the surface normals, thus we can encode the resulting new normals into a new normal texture map which can be mapped onto the original mesh. In this case the detail recovering technique presented by Cignoni et al. [18] and further improved by Sander et al. [19] can be applied to resample a normal map from the refined mesh produced by our normal-enhancement algorithm and the associated set of normal vectors. We can then use common graphics hardware to render the synthesized map very efficiently [20].

Note that this approach could lead to yet another normal enhancement approach: if there exists a *face-continuous* mapping from surface to texture space (e.g. a good texture parameterization of the mesh) then we could build the bitmap representing the normal map of the mesh and then apply on it standard image processing techniques. In most cases, this approach is not feasible because, with some trivial exceptions, there is no simple way to build such a texture parameterization; existing techniques [18,21–23] either build discontinuous mappings or exhibit deformations that make an image processing approach either difficult or not feasible.

## 5. Results

The normal enhancement effect that can be obtained with the application of the formula  $n_E = n + k \cdot (n - n_L)$  depends mainly on two parameters: the amount of low-pass filter that we use to generate the smooth normals  $n_L$  and the value of the weighting constant  $k$  used in the perturbation of the original normal vectors. By tuning these two parameters we can obtain slightly different visual results.

The weighting constant  $k$  affects the *intensity* of the normal enhancement effect; as a rule of thumb we have found, for this parameter, reasonable values in the range [0.2..0.7].

As introduced in Section 4, low-pass filtering is performed by adopting a simple Laplacian kernel: we iteratively average each face normal with the normals of adjacent faces. The number of iterations of this averaging process affects the extent of the smoothing process. By using a large number of smoothing steps the smallest features of a mesh can totally disappear in the smoothed representation and, for this reason, their enhancement can become uniform and, therefore, less detectable. On the other hand, by using a large number



Fig. 9. Some more results on two CAD-like meshes. Again, even simple, plain renderings produce clearer and more informative images when the mesh normals have been enhanced in a preprocessing stage.

of smoothing steps, we obtain a larger extension of the shaded section that can be useful for large features. Fig. 6 shows this situation: (a) is the original mesh, while (b)–(d) show the effect of normal enhancement by using, respectively, 10, 20 and 30 normal averaging iterations. The enhancement is visible in the cross shaped hole: the central section of the vertical wall (see zoom box) is rendered with a slightly lighter shading that almost vanishes in the last two instances (c) and (d). On the other hand, in all cases shown in the figure the shading produced by the proposed technique resolve the shading ambiguity occurring in the original mesh.

Some other examples of the application of the proposed technique are shown in Fig. 7 and 9. In this and the other figures, for each object we show the original and the normal-enhanced model side to side. In both the cases the mesh is rendered using standard

OpenGL shading of the model. For each pair of images, all the rendering parameters (lights, materials, etc.) and model characteristic (number of faces, and tessellation) remain unchanged. Note that, in most cases, the normal enhancement technique effectively resolve many shading ambiguities.

The technique can be applied over irregularly shaped objects, e.g. the Stanford bunny in Fig. 8, rather than the highly regular object we focused on: the result is a sort of high frequency detail enhancement (the fact that the silhouette and the shape of the object is unchanged is hardly notable in this case). Fig. 8 could suggest that the effect of the technique proposed is very similar to what can be obtained by a simpler contrast enhancement (e.g. adopting standard 2D image-processing filters on a rendered image). This is true just in the case of a purely diffuse material lit by a single light source positioned on

the same location of the viewer. Conversely, if we have a more complex lighting environment or the object is rather specular, the intrinsic 3D nature of the enhancement emerges. As an example, see Fig. 8 which shows also some images of a shiny bunny.

Due to the simplicity of the proposed technique, the implementation is straightforward. The expected running time of the normal enhancement and resampling process is usually very low, in the orders of a few seconds for any mesh that can be rendered interactively.

### 5.1. Side effects

Under particular rendering conditions, some enhanced model may suggest an artifact concavity of flat surfaces (for example, see the top-right image in Fig. 7). This effect, however, can be kept under control using low values for either of the two parameters (still getting most of the visual improvements). It is not always the case that this represents a real disadvantage. There are a number of applications where the context clearly determines the object's regularity or planarity, such as mechanical CAD parts, architectural and interior design etc. In all these cases, the effect of our technique can be considered to be some sort of "artistic license". There are many applications (e.g. assembly instructions) where the geometry of the objects in the scene is known, and the focus of the visual presentation is to clarify the respective positions and inter-relations between parts.

On the other hand, it has to be noted that we do not modify the geometry, and thus our approach can be used as a non-permanent modification of the object obtained by just using a different rendering modality, which could be toggled on/off by the user in the inspection of a given object.

## 6. Conclusion

We have presented a technique that enhances the shading and the perception of its features by modifying the normals of an object. This *normal enhancement* technique is done on the mesh in a preprocessing stage; the enhanced normals are integrated into the model either by assigning new normal values per vertexes, or through resampled normal maps. This approach is thus decoupled from the rendering algorithm used to effectively produce the image. The enhanced normals, mapped to the input geometry using a standard texture mapping approach, can then be used into any rendering subsystem that support user-specified normals, or interactive bump mapping.

The technique is especially well suited for regular objects (such as CAD models), but can also be used on any 3D mesh: the enhanced normals, once used in a rendering process, results in images that look more

"sharp" and intuitive in the sense that they support a better perception of the shape of the represented object and present less ambiguities. These synthetic images are not quite realistic, but closely resemble a style commonly used by illustrators for the same category of objects.

## References

- [1] Gooch B, Gooch A. Non-photorealistic rendering. A.K. Peters Ltd.; 2001.
- [2] Curtis CJ, Anderson SE, Seims JE, Fleischer KW, Salesin DH. Computer-generated watercolor. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques (SIGGRAPH'97). ACM Press/Addison-Wesley Publishing Co.; 1997. p. 421–30.
- [3] Salisbury MP, Wong MT, Hughes JF, Salesin DH. Orientable textures for image-based pen-and-ink illustration. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques (SIGGRAPH'97). ACM Press/Addison-Wesley Publishing Co.; 1997. p. 401–6.
- [4] Winkenbach G, Salesin DH. Computer-generated pen-and-ink illustration. In: Proceedings of the 21st annual conference on computer graphics and interactive techniques (SIGGRAPH'94). ACM Press; 1994. p. 91–100.
- [5] Majumder A, Gopi M. Hardware accelerated real time charcoal rendering. In: Proceedings of the second international symposium on non-photorealistic animation and rendering. ACM Press; 2002. p. 59–66.
- [6] Strothotte T, Schlechtweg S. Non photorealistic computer graphics: modeling, rendering and animation. Los Alios: Morgan Kaufmann; 2003.
- [7] Saito T, Takahashi T. Comprehensible rendering of 3-d shapes. In: Proceedings of the 17th annual conference on computer graphics and interactive techniques (SIGGRAPH'90). ACM Press; 1990. p. 197–206.
- [8] Schlechtweg S, Strothotte T. Generating scientific illustrations in electronic books. In: Smart graphics. Papers from the 2000 AAAI Spring Symposium (Stanford, March. 2000). Menlo Park: AAAI Press; 2000. p. 8–15.
- [9] Wanger LR, Ferwerda JA, Greenberg DP. Perceiving spatial relationships in computer-generated images. IEEE Computer Graphics and Applications May 1992;12(3): 44–58.
- [10] Gooch A, Gooch B, Shirley P, Cohen E. A non-photorealistic lighting model for automatic technical illustration. In: Proceedings of the 25th annual conference on computer graphics and interactive techniques (SIGGRAPH'98). ACM Press; 1998. p. 447–52.
- [11] Gooch B, Sloan P-PJ, Gooch A, Shirley P, Riesenfeld R. Interactive technical illustration (color plate S. 220). In: Proceedings of the 1999 symposium on Interactive 3D graphics. ACM Press; 1999. p. 31–8.
- [12] Dooley D, Cohen MF. Automatic illustration of 3D geometric models: Lines. Computer Graphics 1990; 24(2):77–82.
- [13] Raskar R. Hardware support for non-photorealistic rendering. In: Proceedings of the ACM SIGGRAPH/



- EUROGRAPHICS workshop on Graphics hardware. ACM Press; 2001. p. 41–7.
- [14] Taubin G. A signal processing approach to fair surface design. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques (SIGGRAPH'95). ACM Press; 1995. p. 351–8.
- [15] Vollmer J, Mencl R, Müller H. Improved Laplacian smoothing of noisy surface meshes. *Computer Graphics Forum* 1999;18(3):131–8.
- [16] Guskov I, Sweldens W, Schröder P. Multiresolution signal processing for meshes. In: Proceedings of the 26th annual conference on computer graphics and interactive techniques (SIGGRAPH'99). New York: ACM Press/Addison-Wesley Publishing Co.; 1999. p. 325–34.
- [17] Desbrun M, Meyer M, Schröder P, Barr AH. Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of the 26th annual conference on computer graphics and interactive techniques (SIGGRAPH'99). ACM Press/Addison-Wesley Publishing Co.; 1999. p. 317–24.
- [18] Cignoni P, Montani C, Rocchini C, Scopigno R, Tarini M. Preserving attribute values on simplified meshes by resampling detail textures. *The Visual Computer* 1999;15(10):519–39 (preliminary results appeared in *IEEE Visualization'98 Proceedings*).
- [19] Sander PV, Gu X, Gortler SJ, Hoppe H, Snyder J. Silhouette clipping. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques (SIGGRAPH'00). ACM Press/Addison-Wesley Publishing Co.; 2000. p. 327–34.
- [20] Kilgard MJ. Practical and robust bump mapping technique for today's GPU's. In: *Advanced opengl game development*; 2000.
- [21] Maillot J, Yahia H, Verroust A. Interactive texture mapping. In: Proceedings of the 20th annual conference on computer graphics and interactive techniques (SIGGRAPH'93). ACM Press; 1993. p. 27–34.
- [22] Maruya M. Generating texture map from object-surface texture data. *Computer Graphics Forum (Proceeding of Eurographics'95)* 1995;14(3):397–405.
- [23] Soucy M, Godin G, Rioux M. A texture-mapping approach for the compression of colored 3d triangulations. *The Visual Computer* 1996;12(10):503–14.
- [24] Woo M, Neider J, Davis T, Shreiner D. *OpenGL programming guide*, 3rd ed. Reading: Addison-Wesley; 1999.