

Light warping for enhanced surface depiction

Romain Vergne Romain Pacanowski Pascal Barla Xavier Granier Christophe Schlick

INRIA Bordeaux University



Figure 1: Our novel light warping approach enhances surface depiction by locally compressing patterns of reflected lighting. Such a process preserves the overall appearance of 3D objects, as exemplified with these two renderings that use drastically different illuminations. Observe how various surface features are properly enhanced in both settings: sharp features on the face, broad variations around shoulders, and rough details on the torso.

Abstract

Recent research on the human visual system shows that our perception of object shape relies in part on compression and stretching of the reflected lighting environment onto its surface. We use this property to enhance the shape depiction of 3D objects by locally warping the environment lighting around main surface features. Contrary to previous work, which require specific illumination, material characteristics and/or stylization choices, our approach enhances surface shape without impairing the desired appearance.

Thanks to our novel local shape descriptor, salient surface features are explicitly extracted in a view-dependent fashion at various scales without the need of any pre-process. We demonstrate our system on a variety of rendering settings, using object materials ranging from diffuse to glossy, to mirror or refractive, with direct or global illumination, and providing styles that range from photorealistic to non-photorealistic. The warping itself is very fast to compute on modern graphics hardware, enabling real-time performance in direct illumination scenarios.

1 Introduction

The creation of compelling hand-drawn illustrations is an activity that requires skills and time. Looking at the work of scientific illustrators [Wood 1994; Hodges 2003], or the creations of artists such as Norman Rockwell or Burne Hogarth [1991], one is forced to admire the efforts put into the realization of these pieces of artwork. Their challenge is not only to imitate accurately an existing scene though, but also to communicate other characteristics of objects in a visually comprehensible manner.

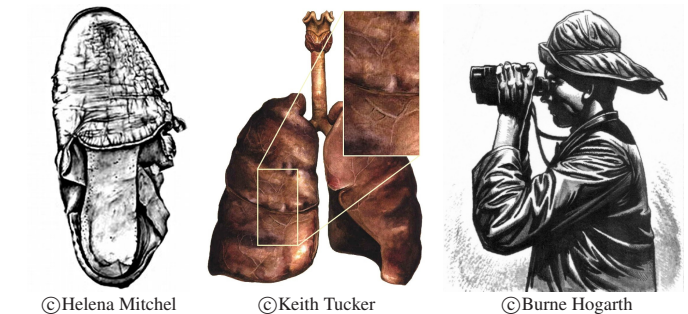


Figure 2: Three examples of artistic surface enhancement with complex materials and illumination. From left to right, a medieval shoe where every little detail has been enhanced; a pair of lungs where small veins are exaggerated; a character wearing shiny cloth exhibiting multiple folds.

Shape is arguably the most important property of objects around us and skilled artists are able to convey shape through the subtle tweaking of shading behaviors. For instance in Archeology, fine surface characteristics are depicted in sharp relief as in Figure 2-left. Medical illustrations such as the lungs in Figure 2-middle often represent anatomical shape and surface details with great accuracy. Note that in both cases, the original material and its appearance under natural lighting conditions are retained. Similar examples may also be found in artistic illustrations, as in Figure 2-right: here, the folds of a garment are skillfully reproduced, while its reflectance characteristics are efficiently conveyed.

Rendering techniques have now reached a point where the simulation of light transport automates the creation of realistic pictures. As such, they offer an invaluable tool to Computer Graphics artists and designers. However, these images still lack the expressive power of many scientific and artistic illustrations. Attempts have been made to visually enhance the shape of 3D objects. Taking inspiration from traditional media, much of the work has focused on Non-Photorealistic Rendering (NPR) techniques. These methods employ specific styles that are efficient at drawing attention to particular surface features of 3D objects. Unfortunately, they also severely restrict the range of possible material and illumination characteristics.

In contrast, this paper investigates the problem of communicating *shape through shading*, yet without impairing the depiction of complex materials and illumination. Our goal is to reproduce some of the aforementioned artistic abilities using rendering techniques. This is not a trivial problem though, as the requirement of faithful simulation of material and illumination seems to leave no degrees of freedom for depicting surface features. Our key idea is to exploit characteristics of the Human Visual System (HVS) to relax these constraints. Indeed, recent work in visual perception has argued that (1) the way the HVS perceives surface shape is highly dependent on view-dependent features (i.e., on the orientation and distance of a surface relative to the point of view [Fleming et al. 2004]). Moreover, it has been shown that (2) the HVS is able to recover such surface features from patterns of reflected lighting [Tarr et al. 1998; Fleming et al. 2004; Ho et al. 2006]. This provides a potential means to enhance shape depiction: in order to better reveal surface features from the current point of view, one can deform patterns of reflected lighting. But an arbitrary deformation may alter the coherence of illumination. Fortunately, evidence has been made that (3) the HVS is relatively insensitive to local inconsistencies in illumination direction [Ostrovsky et al. 2001]. Note that these observations bear some similarities with part of the work of Ramamoorthi et al. [2007].

Our approach takes these three considerations about the HVS into account to enhance surface shape. More precisely, the main contribution of this paper is two-fold. First, we introduce a novel local shape descriptor that extracts *view-centered surface features* at arbitrary scales, improving previous methods in many aspects. Second, we present a new light warping approach that locally *deforms lighting patterns* to reveal important surface features. It preserves material and illumination characteristics, enabling a much wider range of appearances compared to previous work. On the practical side, our approach is also more flexible than previous techniques: it requires no pre-process and works on arbitrary static or dynamic inputs; it is easily incorporated into direct or global illumination renderers, and it adds a relatively small performance overhead, enabling real-time rendering in direct illumination scenarios.

2 Previous work

Line drawing has been an important field of research among previous techniques that tried to address the depiction of object shape. Since the seminal work of Saito and Takahashi [1990], a number of techniques have been proposed, including silhouettes and creases [Nienhaus and Döllner 2004], ridges and valleys [Ohtake et al. 2004], *suggestive contours* [DeCarlo et al. 2003], *apparent ridges* [Judd et al. 2007], *demarcating curves* [Kolomenkin et al. 2008] and *Laplacian lines* [Zhang et al. 2009]. However, no consensus has yet been reached on the “right” set of lines for depicting object shape [Cole et al. 2008].

The abstraction provided by line-based methods is interesting in many respects, because it creates legible pictures with an economy of means. However, this is clearly not always wanted as illustrated in Figure 2, and the focus of our paper is precisely on finding alternatives that preserve material and illumination information, while still efficiently depicting shape. Most line-based methods ignore such properties. A few exceptions include the line drawings of Lee et al. [2007], and the line stylizations of Goodwin et al. [2007] that incorporate shading information. They are restricted to a relatively small subset of materials or illumination effects though.

Another highly popular approach to enhance shape perception is the use of *ambient occlusion* [Pharr and Green 2004]. This method tends to darken surface regions that are less accessible, such as concavities, which is somehow related to accessibility shading approaches [Miller 1994]. Such methods may also depict some sur-

face details, but are best at improving volume appreciation: they will typically ignore many shallow (yet salient) surface details or even smooth them out. Moreover, these methods offer no high-level control to users, and obtaining accurate occlusion information requires time-consuming precomputations that make their use in dynamic scenarios difficult. Furthermore, they are not adaptable to arbitrary illumination, as they correspond to shading effects that typically occur “on a cloudy day” [Langer and Bühlhoff 1999].

The *3D unsharp masking* technique of Ritshel et al. [2008] is also related to such volumetric enhancement of shape, and thus shares the advantages and drawbacks of the aforementioned methods. It increases the contrast of reflected radiance in 3D, and thus enhances indiscriminately geometry, materials and illumination. Although it might be a desired result, as with diffuse materials or cast shadows for instance, it also changes severely the perception of glossy materials as acknowledged by the authors.

Previous work most related to our approach are shading-based techniques that alter reflection rules based on local surface information. They are often inspired from traditional stylized illustrations used in specific domains: Gooch et al. [1998] reproduce effects used in technical illustrations of mechanical parts; *mean-curvature shading* technique by Kindlmann et al. [2003] resembles pen-and-ink archeological illustrations; *normal enhancement* by Cignoni et al. [2005] is inspired from technical pencil drawings; *exaggerated shading* by Rusinkiewicz et al. [2006] uses a set of rules originating from cartography; *geometry-dependent lighting* [Lee et al. 2006] is inspired from anatomical illustration, and *apparent relief* by Vergne et al. [2008] mimics styles found in comics and anime.

All these methods are tailored to a specific combination of style, illumination and material. For instance, Rusinkiewicz et al. [2006] use a very specific cosine shading model unable to accommodate most existing materials or illumination, whereas Vergne et al. [2008] restrict their approach to NPR renderings with a single light and simple materials. Moreover, most of these techniques require an expensive pre-processing step and do not achieve *automatic* levels-of-detail as opposed to ours. A more detailed comparison with most relevant previous work is presented in Section 7.

3 Our approach

The main motivation behind our approach comes from findings of Fleming and Adelson [2004; 2009] who have shown how the perception of curvature from the point of view tends to depend on compressions of reflected light patterns on surface patches. This effect is illustrated in Figure 3, where curved surfaces contract a wider region of the environment lighting than flat surfaces, hence producing more compressed patterns from a particular point of view.

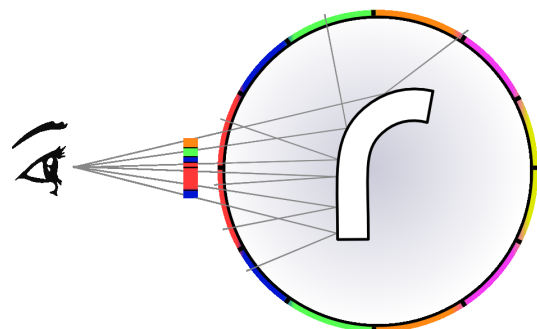


Figure 3: *Compressions of reflected light patterns (here in the mirror case) reveal information about surface curvature from the current point of view. Planar surfaces reflect smaller regions of the environment, whereas curved ones reflect wider regions and are thus more “compressed” for a same surface area. The HVS is likely to use such a cue to estimate curvature.*

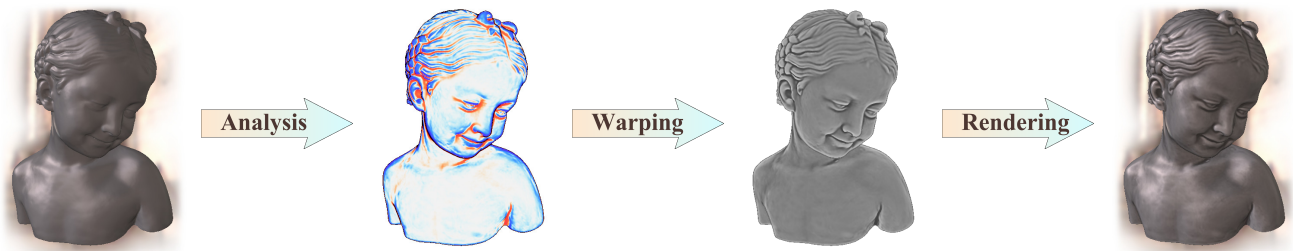


Figure 4: The pipeline of our system. A 3D object in an arbitrary rendering setting is analyzed using its distribution of normals from the current viewpoint. The resulting shape descriptor identifies salient shape features, which are user-selected via high-level controls and mapped to light warping parameters. Finally, the 3D object is then rendered using the locally warped environment lighting and prescribed material characteristics.

Our idea is then to warp incoming lighting at every point in such a way that the compression of reflected light patterns enhances view-dependent surface curvature information. This is done in three stages (see Figure 4):

- **Analysis** (Section 4): We first analyze 3D object surface shape from the current viewpoint. This is done via a novel view-centered *local shape descriptor* that identifies salient surface features at multiple scales.
- **Warping** (Section 5): We then compress or stretch locally the sphere of potential illumination directions to enhance or attenuate surface depiction. Our contribution resides in a view-centered *warping function* that deforms incoming light directions around salient surface features.
- **Rendering** (Section 6): We finally propose different rendering scenarios that incorporate the warped environment lighting. To this end, we propose a *reformulation* of the reflected radiance equation that takes the warped lighting into account.

Every stage of our system is performed in real-time on modern graphics hardware, with the exception of (optional) global illumination routines that are executed off-line.

4 Local shape analysis

The originality of our approach is to analyze surface shape from its normal field in image space. This makes our system very flexible, as normals may either be sampled from 3D surfaces (implicit surfaces, meshes, etc), or read from image-based representations (e.g., RGBN images [Toler-Franklin et al. 2007], normal maps).

4.1 A simple 1D example

To explain the process, let us first study a simple 1D normal field, as shown in the top row of Figure 5. Normals implicitly convey information about the relative depth of points on objects surface. The middle and bottom rows show the first and second derivatives of this depth field respectively. The foremost important features to identify are silhouettes and creases as they represent discontinuities in zeroth- and first-order derivatives respectively. They are important because they represent boundaries between different surface regions, and will thus be treated in a special way in the following. The second most important features are inflection points, as they separate convex from concave regions. Inflection points correspond to extrema of the first-order derivative, and zero-crossings of the second-order derivative.

In between silhouettes, creases and inflections, the magnitude of the second-order derivative gives information about the surface curvature. Note that in our approach, concavities correspond to positive curvature and convexities correspond to negative curvature. We use warm hues for concavities and cold hues for convexities in the remainder of this paper. Our measure of curvature is *view-centered*, as

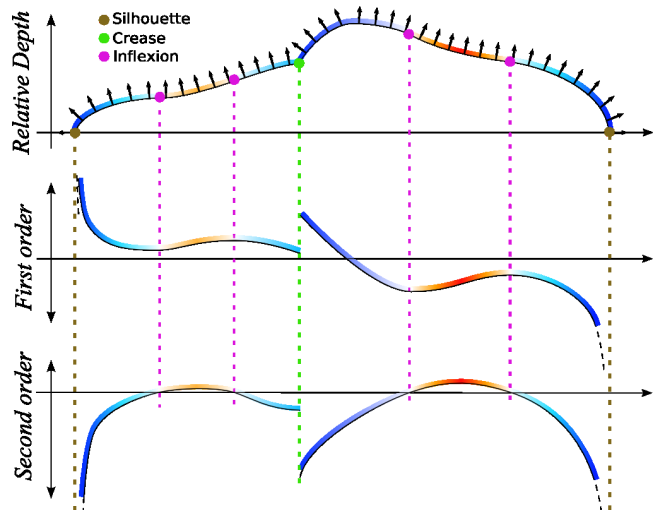


Figure 5: A 1D example: normals implicitly define a height field that represents relative depth information. Differentiating it reveals singularities (silhouettes and creases), as well as concave and convex regions.

it is computed relative to the view direction, as opposed to *object-centered* curvatures that are computed relative to surface normal directions. Previous work essentially made use of object-centered measures. The main advantage of view-centered curvature is that it properly reflects surface foreshortening, as well as the size of projected features. These cues are likely to be taken into account by the HVS, as noted in Section 1. In addition, the view-centered approach has an important number of practical advantages because it is computed dynamically from the current viewpoint.

4.2 Curvature analysis

The analysis is performed similarly in 2D over the whole image. In the following, we denote by \mathbf{x} , \mathbf{y} and \mathbf{z} the axes of image space. As we are only interested in curvature (i.e., second-order information), it is not required to explicitly compute the depth field. Indeed, if we denote the normal at a point p in image space by $\mathbf{n}(p) = (n_x, n_y, n_z)$ and the relative depth by $d(p)$, then there is a direct relationship between the gradient of d and \mathbf{n} , given by:

$$\mathbf{g}(p) = \nabla d(p) = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \begin{pmatrix} -n_x/n_z \\ -n_y/n_z \end{pmatrix}$$

with \mathbf{g} the depth gradient, and d_x and d_y the first-order derivatives of d in the \mathbf{x} and \mathbf{y} directions. In other words, \mathbf{g} is obtained directly from surface normals without having to differentiate d .

At silhouettes, $n_z = 0$ and thus \mathbf{g} is undefined, forbidding any differentiation across them. This makes sense as image neighborhoods should be restricted to connected surface neighborhoods, as

The Hessian of the depth field is then computed by differentiating the gradient:

$$\mathbf{H}(p) = \nabla^T \nabla d(p) = \nabla^T \mathbf{g}(p) = \begin{pmatrix} \mathbf{g}_x & \mathbf{g}_y \end{pmatrix}$$

where \mathbf{g}_x and \mathbf{g}_y are the first-order derivatives of \mathbf{g} in the \mathbf{x} and \mathbf{y} directions. In other words, \mathbf{H} is obtained by differentiating the components of \mathbf{g} . At creases, where \mathbf{g} is not differentiable, \mathbf{H} is undefined, hence differentiation must be restricted across creases as well. However, we mostly consider smooth surfaces in the examples given in the paper and supplemental materials.

\mathbf{H} is a curvature tensor, a symmetric 2×2 matrix that can be easily rewritten as follows:

$$\mathbf{H} = \mathbf{Q}^T \mathbf{DQ} = \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix} \begin{pmatrix} \kappa_u & 0 \\ 0 & \kappa_v \end{pmatrix} \begin{pmatrix} \mathbf{u} & \mathbf{v} \end{pmatrix}^T$$

where κ_u and κ_v are the principal curvatures, and \mathbf{u} and \mathbf{v} correspond to the principal directions.

Our *local shape descriptor* consists of the union of visible silhouettes and creases, and surface points from which we get curvature information via \mathbf{H} . For all examples given in the paper and supplemental materials, we display silhouettes and creases in black, as well as concave and convex regions in warm and cold hues respectively. These features are obtained by remapping mean curvature $H = (\kappa_u + \kappa_v)/2$ on the color scale shown in Figure 6.

4.3 Multi-scale local shape descriptor

A major advantage of using a view-centered curvature tensor for our descriptor is that it confers *automatic* simplification behaviors, as demonstrated in Figure 6-left. Observe how surface features are naturally agglomerated together when the object gets away from the viewpoint. Note that this behavior would be a lot more difficult to obtain with an object-centered description, for instance requiring on-the-fly view-dependent mesh simplification. A more detailed comparison between object- and view-centered curvatures is found in supplemental material.

Another important advantage of a view-centered curvature tensor is that it is easily modified to *dynamically* extract surface features at multiple scales. This is done by integrating (i.e., smoothing) \mathbf{g} over extended neighborhoods in image space. However, as explained in Section 4.1, such neighborhoods must be bounded by silhouettes and creases. For this reason, we perform this integration via anisotropic diffusion [Perona and Malik 1990]:

$$\frac{\partial}{\partial s} \mathbf{g}_s(p) = \nabla \cdot (c(p) \nabla \mathbf{g}_s(p))$$

where s refers to the scale and $c(p)$ is the conductance function that is equal to 0 on silhouettes and creases, and 1 otherwise. We do not reach the steady state of the anisotropic diffusion equation though, but stop at a user-specified number of iterations. It produces a blurred gradient \mathbf{g}_s that preserves silhouettes and creases. The blurred curvature tensor is obtained as before: $\mathbf{H}_s(p) = \nabla^T \mathbf{g}_s$.

The diffusion process is then easily adapted to the creation of levels-of-detail by specifying different amounts of blur in different regions of the picture plane. This is done by letting the user choose an importance function $I(p)$ that controls the number of iterations of the diffusion process: few iterations lead to fine details in important picture regions. Any importance function could be used, and we show an example in Figure 6-right and in the supplemental video.



Figure 6: Left: since the descriptor is computed in image space, it exhibits natural simplification behaviors, whereby coarse scale features only are identified. Right: varying the descriptor’s scale using an importance function (here the focus is on the mouth) creates interesting LOD effects.

4.4 Implementation

In practice, our local shape descriptor is computed per-pixel entirely on the GPU using multiple passes. We take normal and depth buffers as input, and output our descriptor in another buffer. It consists of a pixel-wise multi-scale Hessian \mathbf{H}_s , with silhouette and crease weights w_s and w_c . In the following pseudo-code, p denotes the current pixel and p_i its 3×3 pixel neighborhood.

Algorithm 1 Multi-scale descriptor on the GPU

- 1: $w_s(p) \leftarrow$ Sobel Filter (Depth(p_i))
 - 2: $w_c(p) \leftarrow$ Dihedral Angle ($\mathbf{n}(p_i)$)
 - 3: $\mathbf{g}_0(p) \leftarrow$ Depth Gradient ($\mathbf{n}(p)$)
 - 4: **for** $s \in [1..I(p)]$ **do**
 - 5: $\mathbf{g}_s(p) \leftarrow$ Anisotropic Diffusion ($\mathbf{g}_{s-1}(p_i), w_s(p), w_c(p)$)
 - 6: **end for**
 - 7: $\mathbf{H}_s(p) \leftarrow$ Sobel Filter ($\mathbf{g}_s(p_i)$)
-

There are essentially five steps in the algorithm. Silhouettes (1) are computed as a per-pixel weight w_s using a Sobel filter on depth. We found this approach more accurate and coherent than detecting the locii of image points where $n_z = 0$ in practice. Creases (2) are computed as per-pixel weights w_c as well using the dihedral angle between neighboring normals. The multi-scale depth gradient \mathbf{g}_s is obtained by (3) computing \mathbf{g}_0 and (4-6) discretizing the anisotropic diffusion equation with an iterative solver as explained in [Perona and Malik 1990]. We use $c(p) = 1 - \max(w_s, w_c)$ for the conductance function. Finally, the Hessian (7) is computed by differentiating the multi-scale gradient with a Sobel filter. However, as curvature is not defined at silhouettes and creases, we linearly interpolate between $\mathbf{H}_s(p)$ and a 2×2 matrix of zeros (corresponding to a planar region) using $c(p)$ at these singular locations.

From a practical point of view, our solution offers important advantages. Since it only requires per-pixel normals and depths, it requires no pre-process and works with dynamic 3D scenes. For the same reason, it may be applied to virtually any kind of input data like objects having bump or normal maps, point splat surfaces or RGBN images (we use the occlusion map for silhouette weights in the latter case), as shown in supplemental material. In terms of performance, it is *output-sensitive*, and its complexity is linear in the number of diffusion iterations. In the next section, we show how we make use of the information made available by our local shape analysis with a novel light warping approach.

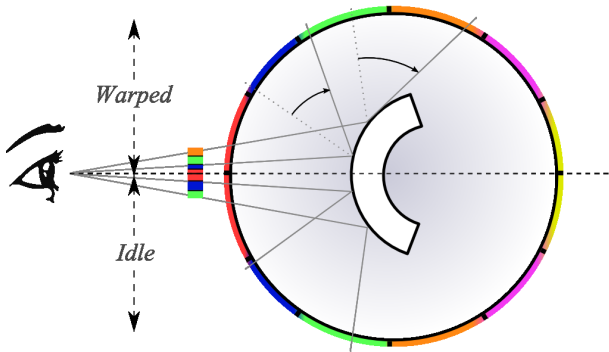


Figure 7: We enhance the curvature information revealed through reflected light patterns by expanding the region that is reflected off the surface.

5 Light Warping

As illustrated in Figure 7, the key idea of the warping approach is to exaggerate the deformation of reflection patterns that naturally occur on curved objects (recall Figure 3). In the simple case of a mirror reflection, one observes that changing the direction of incoming light rays has the effect of contracting a wider region of the environment, as if the object were more curved. An animated example with a glossy object is shown in the supplemental video.

We require the warping function to be a bijective mapping in the sphere of directions, so that the inverse warping function is analytically defined. Moreover, as noted in [Fleming et al. 2009], the compression of reflected light patterns reflects the anisotropy of curvature, defined as the ratio of principal curvatures. As this is likely to be a salient shape cue for the HVS, we design the warping function so that it deforms incoming illumination in different ways along principal curvature directions \mathbf{u} and \mathbf{v} . Since our descriptor provides curvature information in the form of a symmetric tensor in image space, we also require our warping function to be symmetric with respect to \mathbf{u} and \mathbf{v} , and to leave \mathbf{z} invariant. Every light direction is thus transformed into the $\{\mathbf{u}, \mathbf{v}, \mathbf{z}\}$ reference frame prior to warping, and transformed back afterwards.

Curvature and light directions are not expressed in the same coordinate system though. In order to establish correspondences between cartesian and angular spaces, we use a stereographic projection on the image plane. The process is illustrated in Figure 8: (1) the light direction ℓ is stereographically projected on the plane $\mathbf{z} = 1$ to give $\bar{\ell} = S(\ell)$; (2) $\bar{\ell}$ is warped according to curvature information, yielding $\bar{\ell}' = W_S(\bar{\ell})$; (3) $\bar{\ell}'$ is mapped back to the sphere of directions to give the warped light direction $\ell' = S^{-1}(\bar{\ell}')$.

Given a light direction $\ell = (\ell_u, \ell_v, \ell_z)$, the stereographic projection S is defined by:

$$S(\ell) = (a, b, c) = \left(\frac{2\ell_u}{\ell_z + 1}, \frac{2\ell_v}{\ell_z + 1}, 1 \right).$$

The $(0, 0, 1)$ direction is projected on the origin of the stereographic plane. Intermediate light directions are projected further from the origin as they get closer to the $(0, 0, -1)$ direction, which is projected to infinity.

The warping function is simply defined as a curvature-dependent non-linear scaling on the stereographic plane (see Figure 8-right):

$$W_S(\bar{\ell}) = (a', b', c') = (\lambda_u a, \lambda_v b, 1).$$

The scaling factors $\lambda_{u|v}$, are computed by mapping the local curvatures $\kappa_{u|v}$ into an angular deviation on the sphere.

In our implementation, we use $\lambda_{u|v} = \tan(\arctan(\alpha \kappa_{u|v})/6 + \pi/4)$. It guarantees that at most one half of the lighting energy found on

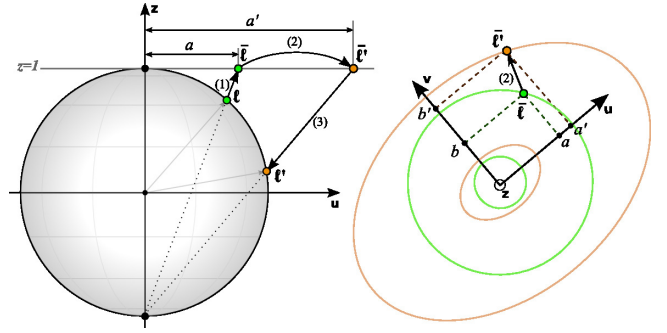


Figure 8: Left: a 1D illustration of the warping process. The light direction is (1) projected stereographically, (2) warped to a new position in stereographic space and (3) projected back to the sphere of directions. Right: an illustration of step (2) in 2D. Note the symmetry around \mathbf{u} and \mathbf{v} .

one side of the hemisphere of directions is warped to the other one. In this formulation, α is a user-defined parameter that controls the amount of warping performed according to the curvature, while the anisotropy of curvature is naturally taken into account.

The inverse stereographic projection is given by:

$$S^{-1}(\bar{\ell}) = (\ell'_u, \ell'_v, \ell'_z) = (a't, b't, 2t - 1)$$

where $t = 4/(4 + a'^2 + b'^2)$ describes the parametric location of the intersection between the sphere and the projection direction.

We concatenate these operations into a single warping function $W = S^{-1} \circ W_S \circ S$, yielding:

$$W(\ell) = \left(\frac{2t\lambda_u\ell_u}{1+\ell_z}, \frac{2t\lambda_v\ell_v}{1+\ell_z}, 2t-1 \right), \quad t = \frac{(1+\ell_z)^2}{(1+\ell_z)^2 + \lambda_u^2\ell_u^2 + \lambda_v^2\ell_v^2}.$$

Note that the inverse warping function $W^{-1} = S^{-1} \circ W_S^{-1} \circ S$ is simply obtained by using the inverse of λ_u and λ_v , (i.e., by replacing α by $-\alpha$).

6 Rendering results

The final stage in our system is to render 3D objects with arbitrary materials and illumination, while taking into account the way the environment lighting must be warped at each surface point. We illustrate this approach with photorealistic as well as non-photorealistic scenarios, with both real-time and off-line renderers.

6.1 Photorealistic rendering

We first reformulate the reflected radiance equation to take the light warping into account:

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega} \rho(\mathbf{e}, \ell) \langle \mathbf{n} \cdot \ell \rangle L(\mathbf{p} \leftarrow W(\ell)) d\ell \quad (1)$$

where \mathbf{p} is the surface point, \mathbf{e} is the viewpoint direction, ℓ is the incoming lighting direction, Ω is the sphere of directions, ρ is the BRDF and W is the warping function as defined in Section 5. We clamp light directions (both original and warped) to the hemisphere of directions around \mathbf{n} , in a way similar to the clamping done when using bump or normal maps.

The discretization of Equation 1 may raise performance and quality issues though. Indeed, it is common to sample light sources in pre-process to reduce noise in the results (e.g., Krivánek and Colbert [2008]). However, since the light warping is *different at every point*, such approaches become intractable with Equation 1. To enable pre-sampling of light sources, we re-write L' by substituting $\ell' = W(\ell)$ to ℓ :

$$L'(\mathbf{p} \rightarrow \mathbf{e}) = \int_{\Omega} \rho(\mathbf{e}, W^{-1}(\ell')) \langle \mathbf{n} \cdot W^{-1}(\ell') \rangle L(\mathbf{p} \leftarrow \ell') J d\ell' \quad (2)$$

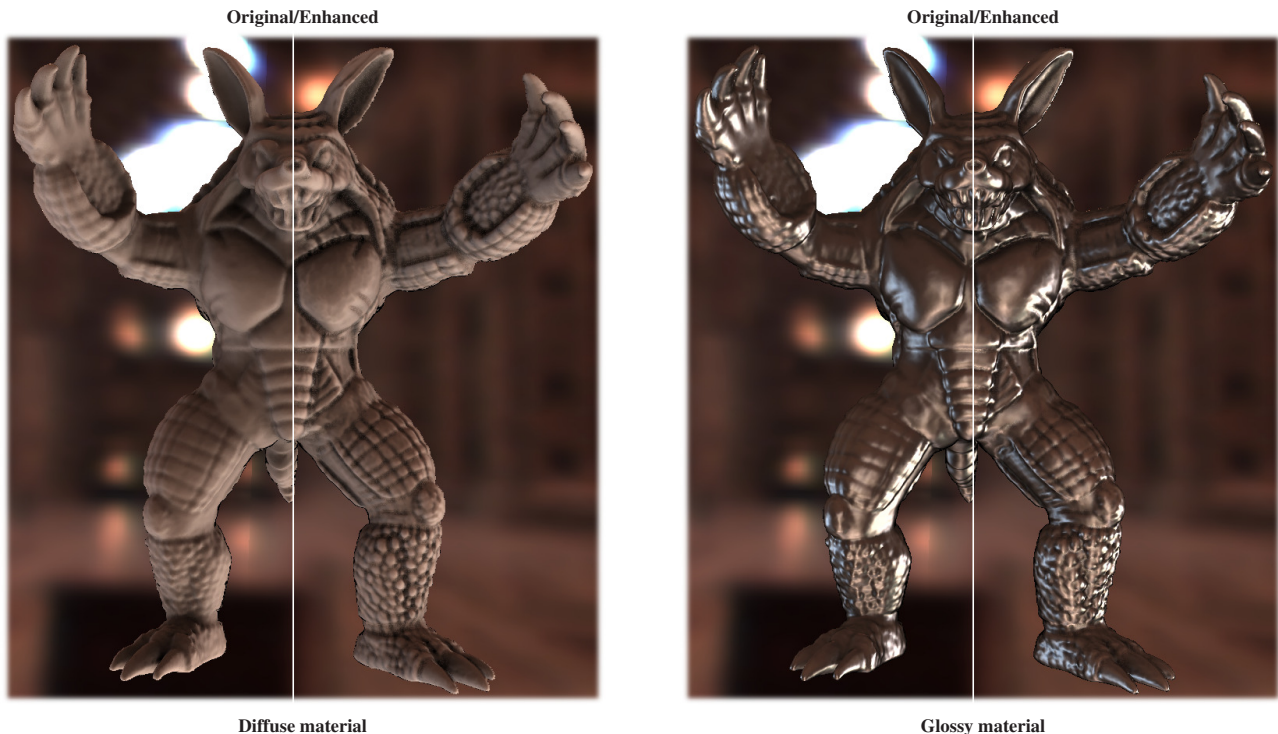


Figure 9: Photorealistic rendering results: the Armadillo model rendered with diffuse and glossy materials. Each side of the figure shows original and warped lighting results. Note how surface features are consistently enhanced in all cases without having to modify warping parameters.

where J is the jacobian of W^{-1} (see supplemental materials):

$$J = \frac{4\lambda_u^3 \lambda_v^3 (1 + \ell_z')^2}{(\lambda_u^2 \lambda_v^2 (1 + \ell_z')^2 + \lambda_v^2 \ell_u'^2 + \lambda_u^2 \ell_v'^2)^2}$$

We implemented the light warping approach in different renderers. In both cases, we used Ashikhmin’s BRDF model [Ashikhmin et al. 2000]. Our real-time rendering system evaluates Equation 2 using pre-sampled environment lights; however, it avoids computing visibility information and ignores indirect illumination. Figure 1 shows how the shape of an input 3D object is enhanced in two different illumination settings with this system. Note how the enhancement remains coherent while the patterns of reflected lighting are completely different in each image; indeed, the only cue we provide here is the deformation of patterns. Additional real-time captures using this rendering system are shown in the supplemental video.

In our off-line rendering system, we compute full global illumination results, with Equation 1 for indirect lighting and Equation 2 for direct lighting, using our path tracer [Dutr e et al. 2006], applying light warping only to the *first* ray bounce. We also implemented a warped ambient occlusion used with diffuse materials. After being warped, light rays have a different visibility; this change of visibility enhances shape as is best seen in supplemental results. Renderings are shown in Figure 9, where the shape of the same input 3D object is enhanced in each configuration. Again, surface features are enhanced no matter the material characteristics. We also experimented with purely reflective and refractive materials as shown in supplemental materials.

The complexity of our light warping approach is linear in the number of sampled light directions. In practice, applying the warping function is negligible with global illumination, but decreases frame rate by 50% with direct illumination. However, we still get real-time frame rates in practice: for instance, the results in Figure 1 are obtained at 37 fps in 800×600 using 54 lights.

6.2 Non-photorealistic rendering

Finally, we experimented with non-photorealistic rendering techniques using our light warping approach. In order to *exaggerate* the enhancement obtained by light warping, we incorporate a curvature-dependent contrast enhancement. The exaggerated reflected radiance is then given by

$$L'_\gamma(\mathbf{p} \rightarrow \mathbf{e}) = (\lambda_u \lambda_v)^\gamma L'(\mathbf{p} \rightarrow \mathbf{e}) \quad (3)$$

where $\gamma \in [-1, 1]$ is a contrast parameter. When both $\kappa_u = 0$ and $\kappa_v = 0$, $L'_\gamma = L'$; in other cases, contrast is increased depending on curvature and warping magnitudes. When applied to an object with diffuse material and minimal illumination, this method comes close to the mean curvature shading technique [Kindlmann et al. 2003].

Figure 10 shows the effect of using Equation 3 in our real-time renderer with both natural and minimal illumination. We also applied a stylized quantization algorithm [Winnem oller et al. 2006] to both renderings that shows how the very same warped lighting is able to enhance stylized shading. The end result is a compelling cartoon style that works with arbitrary materials and illumination.

7 Discussion

Our local shape descriptor bears some similarities with the work of Judd et al. [2007] and Vergne et al. [2008] who have investigated view-dependent approaches to shape depiction in the past. However, they provide only partial curvature information: either the maximum principal curvature in [Judd et al. 2007], or a blending between object- and view-centered curvatures in [Vergne et al. 2008]. In the former case, it limits the method to line-based renderings, while in the latter case, it results in objectionable artifacts around silhouettes. Our method is considerably simpler: it requires no pre-process in object space and basically consists in applying filtering operations to normal and depth buffers in the picture plane.

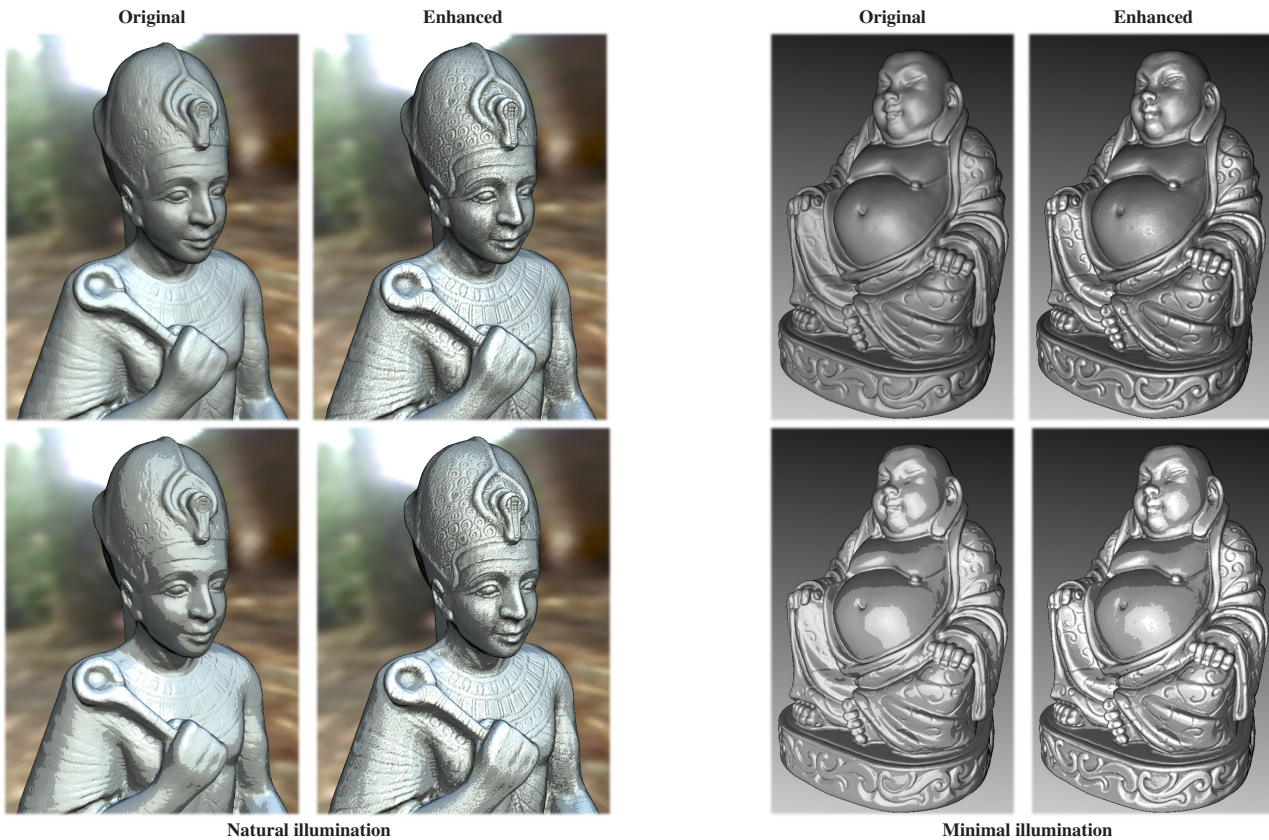


Figure 10: Non-photorealistic rendering results. Top row: light warping with increased contrast confers an even more exaggerated look in natural as well as minimal illumination settings. Bottom row: the enhancement is retained when quantifying the results to give a cartoon appearance.

When getting close to a surface mesh, our descriptor starts enhancing geometry tessellation, as shown in the inset image (zoomed from Figure 6). This may be seen as a limitation, but it is no surprise as normals are only C^0 continuous across triangle edges due to Phong interpolation. The simplest way to address this issue is to use dynamically subdivided meshes or implicit surfaces for instance. We also believe that adapting the descriptor’s scale based on surface depth could smooth areas of coarse tessellation.

The light warping approach we introduced enhances surface features with arbitrary materials, illuminations and styles. It may produce results similar to exaggerated shading or 3D unsharp masking as shown in Figure 11. Compared to [Rusinkiewicz et al. 2006], it enhances surface shape with a much wider range of materials. Furthermore, exaggerated shading suffers from light direction sensitivity, and tends to flatten the overall shape perception, as shown in the supplemental video. Besides, it requires a time-consuming pre-process and yet does not incorporate *automatic* simplification behaviors as opposed to our approach. Compared to [Ritschel et al. 2008], our system offers a greater control as it enhances the surface features *uniformly*, while 3D unsharp masking increases indiscriminately radiance contrast, resulting in irregular enhancement and alteration of material properties. Our approach is also simpler to control compared to previous work, as it offers 3 intuitive parameters: warping magnitude α , lighting contrast γ , and feature scale s .

The light warping technique shows some limitations though. First, it depends on the existence of lighting variations in the scene; this appears to be related to the statistics of natural environments [Fleming et al. 2009]. In practice, it is always possible to enhance surface shape using Equation 3 in cases where the environment lighting has

few variations. Second, it reaches its limits with pure reflections and refractions on objects exhibiting many surface details, because it tends to make the picture less legible as a whole. The overall shape of cast shadows may also be distorted to favor the depiction of surface features. A better balance between surface and shadow shape depiction might thus be needed. Moreover, our warping function tends to sharpen shading transitions when α is pushed to high values, hence affecting material perception. Finally, warping increases noise in off-line renderings and adds a relatively small overhead in real-time renderings. Concerning the noise issue, we plan to investigate specific anti-aliasing methods in the future. Extending our approach to pre-computed warped radiance transfer would be an interesting solution to increase performance.

8 Conclusion and future work

We have presented a new approach to surface shape enhancement called light warping that preserves material and illumination characteristics as well as stylistic choices. It also has a number of practical advantages over previous methods such as flexibility with respect to input data representations, automatic as well as controllable levels-of-detail, and real-time rendering on the GPU. In future work, we plan to exploit the properties of our local shape descriptor for producing line-based renderings in various styles, as we believe it exhibits most of the surface features needed to create rich line drawings. Moreover, we presented one way of performing light warping in stereographic space, but we would like to investigate other potential functions. In particular, we could imagine making use of additional information such as an explicit description of the environment illumination. Finally, an interesting direction of research would be to study the connections of our local shape descriptor and light warping technique with visual perception.

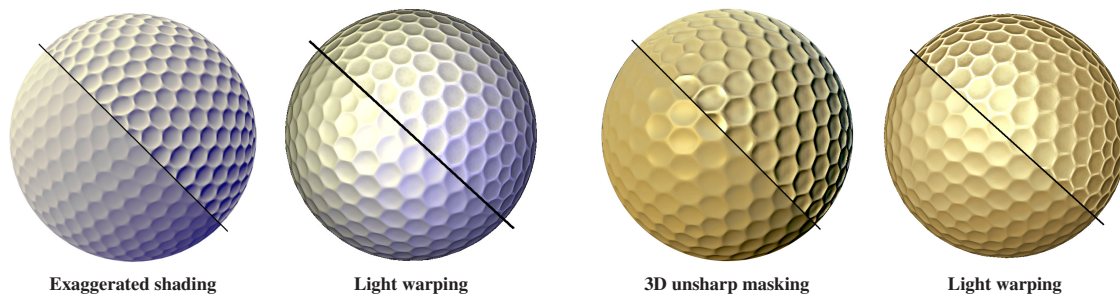


Figure 11: Contrary to other approaches, light warping is able to properly enhance surface details with non-diffuse materials (using Equation 3 here). Left: exaggerated shading is limited to cosine shading. Right: 3D unsharp masking alters material appearance and enhances surface details in a non-uniform way. Images of previous techniques have been extracted from original papers and supplemental materials.

Acknowledgement

We thank the members of the IPARLA team and Roland Fleming for their useful feedback, and Maitena Vives for mentioning the work of Norman Rockwell. We are grateful to the Aim@shape library for 3D models and Paul Debevec for environment maps. This work has been sponsored by the ANR-08-JCJC-0078-01 project.

References

- ASHIKHMIN, M., PREMOZE, S., AND SHIRLEY, P. 2000. A microfacet-based BRDF generator. In *Proc. ACM SIGGRAPH '00*, ACM, 65–74.
- CIGNONI, P., SCOPIGNO, R., AND TARINI, M. 2005. A simple Normal Enhancement technique for Interactive Non-photorealistic Renderings. *Comp. & Graph.* 29, 1, 125–133.
- COLE, F., GOLOVINSKIY, A., LIMPAECHER, A., BARRROS, H. S., FINKELSTEIN, A., FUNKHOUSER, T., AND RUSINKIEWICZ, S. 2008. Where Do People Draw Lines? *ACM Trans. Graph. (Proc. SIGGRAPH 2008)* 27, 3, 1–11.
- DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive Contours for Conveying Shape. *ACM Trans. Graph. (Proc. SIGGRAPH 2003)* 22, 3 (July), 848–855.
- DUTRÉ, P., BALA, K., AND BEKAERT, P. 2006. *Advanced Global Illumination (Second Edition)*. A. K. Peters, Ltd.
- FLEMING, R. W., TORRALBA, A., AND ADELSON, E. H. 2004. Specular reflections and the perception of shape. *J. Vis.* 4, 9 (9), 798–820.
- FLEMING, R. W., TORRALBA, A., AND ADELSON, E. H. 2009. *Three dimensional shape perception*. Springer Verlag, ch. Shape from sheen, to appear.
- GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A Non-Photorealistic Lighting Model For Automatic Technical Illustration. In *Proc. ACM SIGGRAPH '98*, ACM, 447–452.
- GOODWIN, T., VOLLIK, I., AND HERTZMANN, A. 2007. Isophote distance: a shading approach to artistic stroke thickness. In *NPAP '07: Proc. international symposium on Non-photorealistic animation and rendering*, ACM, 53–62.
- HO, Y.-X., LANDY, M. S., AND MALONEY, L. T. 2006. How direction of illumination affects visually perceived surface roughness. *J. Vis.* 6, 5 (5), 634–648.
- HODGES, E. R. S. 2003. *The Guild Handbook of Scientific Illustration*. Wiley.
- HOGARTH, B. 1991. *Dynamic Light and Shade*. Watson Guptill.
- JUDD, T., DURAND, F., AND ADELSON, E. H. 2007. Apparent Ridges for Line Drawing. *ACM Trans. Graph. (Proc. SIGGRAPH 2007)* 26, 3, 19.
- KINDLMANN, G., WHITAKER, R., TASDIZEN, T., AND MÖLLER, T. 2003. Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications. In *Proc. IEEE Visualization 2003*, 513–520.
- KOLOMENKIN, M., SHIMSHONI, I., AND TAL, A. 2008. Demarcating Curves for Shape Illustration. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2008)* 27, 5, 1–9.
- KRIVÁNEK, J., AND COLBERT, M. 2008. Real-time shading with filtered importance sampling. *Comp. Graph. Forum (Proc. EUROGRAPHICS Symposium on Rendering 2008)* 27, 4.
- LANGER, M., AND BÜLTHOFF, H. H. 1999. Perception of shape from shading on a cloudy day. Tech. Rep. 73, Tübingen, Germany, oct.
- LEE, C. H., HAO, X., AND VARSHNEY, A. 2006. Geometry-dependent lighting. *IEEE Transactions on Visualization and Computer Graphics* 12, 2, 197–207.
- LEE, Y., MARKOSIAN, L., LEE, S., AND HUGHES, J. F. 2007. Line drawings via abstracted shading. *ACM Trans. Graph.* 26, 3, 18.
- MILLER, G. 1994. Efficient Algorithms for Local and Global Accessibility Shading. In *Proc. ACM SIGGRAPH '94*, ACM, 319–326.
- NIENHAUS, M., AND DÖLLNER, J. 2004. Blueprints: illustrating architecture and technical parts using hardware-accelerated non-photorealistic rendering. In *Graphics Interface (GI'04)*, Canadian Human-Computer Communications Society, 49–56.
- OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Trans. Graph. (Proc. SIGGRAPH 2004)* 23, 23, 609–612.
- OSTROVSKY, Y., CAVANAGH, P., AND SINHA, P. 2001. Perceiving Illumination Inconsistencies in Scenes. In *MIT AIM*.
- PERONA, P., AND MALIK, J. 1990. Scale-Space and Edge Detection Using Anisotropic Diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 7 (July), 629–639.
- PHARR, M., AND GREEN, S. 2004. *GPU Gems*. Addison-Wesley, ch. Ambient Occlusion.
- RAMAMOORTHY, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A First-Order Analysis of Lighting, Shading, and Shadows. *ACM Trans. Graph.* 26, 1, 2.
- RITSCHEL, T., SMITH, K., IHRKE, M., GROSCH, T., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2008. 3D Unsharp Masking for Scene Coherent Enhancement. *ACM Trans. Graph. (Proc. SIGGRAPH 2008)* 27, 3, 1–8.
- RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated Shading for Depicting Shape and Detail. *ACM Trans. Graph. (Proc. SIGGRAPH 2006)* 25, 3, 1199–1205.
- SAITO, T., AND TAKAHASHI, T. 1990. Comprehensible Rendering of 3-D Shapes. In *Proc. ACM SIGGRAPH '90*, ACM, 197–206.
- TARR, M. J., KERSTEN, D., AND BÜLTHOFF, H. H. 1998. Why the visual recognition system might encode the effects of illumination. *Vision Research* 28, 2259–2275.
- TOLER-FRANKLIN, C., FINKELSTEIN, A., AND RUSINKIEWICZ, S. 2007. Illustration of Complex Real-World Objects using Images with Normals. In *NPAP '07: Proc. international symposium on Non-photorealistic animation and rendering*, ACM, 111–119.
- VERGNE, R., BARLA, P., GRANIER, X., AND SCHLICK, C. 2008. Apparent relief: a shape descriptor for stylized shading. In *NPAP '08: Proc. international symposium on Non-photorealistic animation and rendering*, ACM, 23–29.
- WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. *ACM Trans. Graph. (Proc. SIGGRAPH 2006)* 25, 3, 1221–1226.
- WOOD, P. 1994. *Scientific Illustration: A Guide to Biological, Zoological, and Medical Rendering Techniques, Design, Printing, and Display*, 2nd ed. John Wiley and Sons, Inc., New York.
- ZHANG, L., HE, Y., XIE, X., AND CHEN, W. 2009. Laplacian Lines for Real Time Shape Illustration. In *ISD '09: Proc. symposium on Interactive 3D graphics and games*, ACM.