Stylized Depiction of Images with Normals

Corey Toler-Franklin Adam Finkelstein Szymon Rusinkiewicz
Princeton University

Figure 1: *From photographs of an object under varying lighting (one example is shown at left), we infer albedo and normals (second image). We describe how to use these to produce abstracted illustrations in a variety of styles based on toon shading, lines, curvature shading, and exaggerated shading.*

Abstract

This paper investigates the creation of non-photorealistic illustrations from a type of data lying between simple 2D images and full 3D models: images with both a color (albedo) and a surface normal stored at each pixel. Images with normals combine an acquisition process only mildly more complex than that for digital photographs (and significantly easier than 3D scanning) with the power and flexibility of tools similar to those originally developed for full 3D models. We investigate methods for signal processing on images with normals, developing algorithms for scale-space analysis, derivative (i.e., curvature) estimation, and segmentation. These are used to implement analogues of stylized rendering techniques such as toon shading, line drawing, curvature shading, and exaggerated shading. We show that our rendering pipeline can produce detailed yet understandable illustrations in medical, technical, and archaeological domains.

Keywords: Non-photorealistic rendering, shape depiction

1 Introduction

One of the greatest strengths of computer graphics is the stylistic flexibility it affords the designer. The same scene description, consisting of 3D shape, illumination, and reflectance (texture), can be either rendered photorealistically or depicted in styles that emphasize and highlight particular features while abstracting or deemphasizing others. For this reason, in recent years there has been an increased interest in developing methods for producing illustrations, diagrams, and visualizations that start out as 3D models.

Generating such visualizations from 3D models is typically more powerful than starting with 2D images, since image manipulation cannot distinguish whether intensity variations are caused by shape, illumination, or reflectance. However, the traditional difficulty of working with 3D objects and scenes lies in acquisition: it is time-consuming and expensive to capture complete models of objects, let alone scenes. Most technologies for 3D acquisition capture only part of an object at a time, and reconstructing a complete model may require acquiring, registering, and merging dozens to hundreds of scans, especially for objects with high geometric complexity. While there has been recent work on improving the practicality of range-scanning, 3D acquisition continues to require significantly greater effort than does digital photography.

This paper investigates the creation of illustrations from a type of data lying between simple 2D images and full 3D models: images with a surface normal stored at each pixel. Such datasets may be captured using *photometric stereo* [Woodham 1980], a form of shape-from-shading in which normals are inferred from images of an object illuminated from several different directions and captured from a single camera position. Such datasets have been acquired in previous projects, such as IBM's digitization of Michelangelo's Florentine Pietà [Bernardini et al. 2002], and researchers including Debevec et al. [2000] have demonstrated that it is possible to capture images with normals at sufficiently high quality to be used for cinematic relighting.

We believe that images with normals or "RGBN images" may become an important and widely-used data type, because of the ease, flexibility, and quality with which they may be acquired, and because they contain enough information to permit sophisticated analysis and depiction. In short, they combine an acquisition process only mildly more complex than that for digital photographs with the power and flexibility of tools similar to those originally developed for full 3D models (with the obvious and important limitations that RGBN images do not allow for easy change of viewpoint or realistic cast shadows).

In this paper we build upon previous work on *realistic* relighting to investigate the generation of *stylized* renderings from RGBN images. We first develop tools for segmentation and signal processing on RGBN images (Section 4), including smoothing and derivative estimation. Next, we investigate rendering styles (Section 5) such as toon shading, line drawing, curvature shading, and exaggerated

shading. We show that our rendering pipeline can produce detailed yet understandable illustrations for technical, medical, and archaeological domains, and for any task requiring clear and unambiguous exploration and communication of shape and detail.

2 Previous Work

Non-photorealistic rendering (NPR) refers to a class of techniques that simulate natural artistic media or include non-physical abstraction, exaggeration, or iconic depiction. Most existing NPR methods begin with either a single source image or a 3D model. While the former category includes image analysis, filtering, and simulation algorithms capable of reproducing effects such as pen-and-ink drawings, oil paintings, or watercolors, these methods are limited by an inability to distinguish between variations in pixel intensity due to reflectance, geometry, and lighting. Thus, more recent NPR algorithms, capable of more meaningful abstraction and stylization, typically operate on 3D models and scenes. This paper considers two classes of NPR effects previously explored for 3D models: enhanced shading/lighting models and shape-conveying lines.

NPR shading models are often simply functions of the surface normal and light direction that result in effects such as toon shading [Decaudin 1996], warm-to-cool transitions [Gooch et al. 1998], cartographic hill shading [Horn 1981], or other artist-specified effects [Sloan et al. 2001]. More complex models are also possible, with some systems employing curvature-based shading to emphasize creases [Kindlmann et al. 2003] and others using light positions carefully selected to increase local contrast [Lee et al. 2006]. The "exaggerated shading" technique increases contrast across scales and for all surface orientations [Rusinkiewicz et al. 2006].

The second class of NPR effects focuses on "sparse," shape-conveying linear features (as opposed to tone-conveying lines such as hatch strokes). The most frequently used are: (1) silhouettes and occluding contours (interior and exterior silhouettes — locations at which the surface normal is perpendicular to the view direction) [Dooley and Cohen 1990; Elber and Cohen 1990; Winkenbach and Salesin 1994; Markosian et al. 1997; Hertzmann and Zorin 2000]; and (2) ridge and valley lines (local maxima of principal curvature magnitude in a principal direction) [Interrante et al. 1995; Thirion and Gourdon 1996; Pauly et al. 2003; Ohtake et al. 2004]. Another type of line we consider is suggestive contours (locations at which occluding contours appear with minimal change in viewpoint) [DeCarlo et al. 2003; DeCarlo et al. 2004], which complement occluding contours to better depict shape.

Although some shading models, such as depth cueing and depth shading [Rheingans and Ebert 2001; Cohen et al. 2004], operate on surface positions, most require knowing only the surface normals and perhaps higher-order surface derivatives (such as curvature). Similarly, the lines considered above may all be defined with reference only to the surface normals, curvatures, and derivatives of curvature. This paper develops a signal processing theory that correctly and consistently handles operations such as smoothing and finding derivatives of all orders (Section 4). We implement many of the above shading and line-drawing methods for RGBN images (Section 5), and also address the question of how to combine shape-based stylization with surface color information in a way that is clear and easy to understand.

Depiction of Images with Discontinuities: An important class of nonphotorealistic rendering algorithms that operate on more than simply colors consists of techniques that attempt to depict depth discontinuities (which may be located in synthetic renderings [Saito and Takahashi 1990] or in real-world datasets [Raskar et al. 2004]). Typically, these methods produce "haloing" or "shadowing" effects around depth discontinuities [Rheingans and Ebert 2001; Raskar et al. 2004; Tan et al. 2004; Luft et al. 2006], leading to a better perception of relative depth. We describe a similar shadowing effect in this paper, using depth discontinuities detected by the method of Raskar et al. [2004].

Rendering of Images with Normals: While some applications of normal maps recovered from real objects have integrated them or combined them with coarse geometry, there have been several applications that retained the normals in the form of an image, using them for rendering. One class involves relighting: an image under arbitrary new lighting is formed by computing a (local) lighting model per-pixel [Debevec et al. 2000]. This is conceptually similar to *deferred shading* for rasterization [Deering et al. 1988], in which a normal is computed per fragment and final colors are computed in a separate pass. The rendering need not be photorealistic: some work has explored enhancing subtle detail by *reflectance transformation*, i.e. using a (usually highly-specular) material that emphasizes deviations in the normals [Malzbender et al. 2001; Wenger et al. 2005; Malzbender et al. 2006]. This approach has been successful in domains such as archaeology and art history [Mudge et al. 2005; Mudge et al. 2006]. Bartesaghi et al. [2005] have explored a style based on hatching, with the density of hatch marks controlled by image intensity while their direction follows principal direction fields extracted from stereo or photometric stereo. The present paper builds upon the specific rendering algorithms considered in previous work by developing a suite of geometric processing tools for RGBN images, then applying them to a variety of NPR styles.

3 Acquisition

While the tools we develop for rendering with RGBN images may use datasets obtained using any technique, all data used in this paper were produced by photometric stereo. We briefly describe the technique, as well as the acquisition apparatus used for most of the illustrations in this paper.

Background on Photometric Stereo: Systems based on the principle of shape from shading [Horn 1970] are the primary methods for acquiring high-quality, dense normal maps. In particular, several recent systems have used variants of photometric stereo [Woodham 1980; Rushmeier and Bernardini 1999], which recovers normals under the assumptions of Lambertian (perfect diffuse) shading and multiple light sources of known position and brightness. While some recent research has concentrated on relaxing these requirements, with work on non-Lambertian surfaces [Tagare and deFigueiredo 1991; Georghiades 2003] and unknown distant lighting [Basri and Jacobs 2001], we use only the most basic technique.

The photometric stereo calculation operates on each pixel independently. A surface point of albedo a and normal $\hat{n}$ is illuminated multiple times by light sources with directions $\hat{l}_i$, yielding intensities e_i. Using the Lambertian lighting law

$$e_i = a\,(\hat{n} \cdot \hat{l}_i), \tag{1}$$

we may arrange all the available data into a matrix equation:

$$\begin{pmatrix} l_{1,x} & l_{1,y} & l_{1,z} \\ l_{2,x} & l_{2,y} & l_{2,z} \\ l_{3,x} & l_{3,y} & l_{3,z} \\ & \vdots & \end{pmatrix} \begin{pmatrix} a\,n_x \\ a\,n_y \\ a\,n_z \end{pmatrix} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \end{pmatrix}. \tag{2}$$

This equation may be solved using least squares for the vector $a\hat{n}$, whose length and direction give a and $\hat{n}$, respectively. Heuristics are used to exclude shadows and specularities from the calculation. Typically, at least 5 or 6 lights are used (rather than the minimum 3) to provide noise reduction and allow for specularity and shadow rejection. Domes with tens of light sources have been used for greater robustness [Wenger et al. 2005; Weyrich et al. 2006], as have manual and computerized rigs that allow for dozens or hundreds of light positions [Debevec et al. 2000; Malzbender et al. 2001].

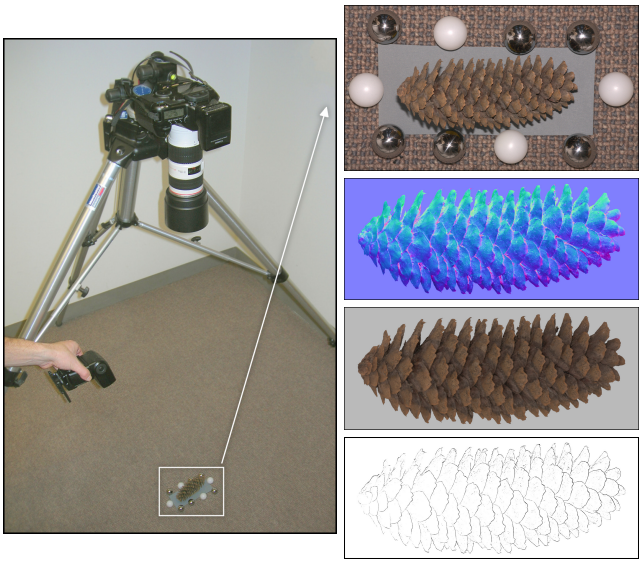


Figure 2: Left: *our setup for capturing RGBN images, using a digital SLR camera and hand-held flash. White and mirror spheres are used to find the flash intensity and position for each captured image.* ***Right:*** *An original image, together with extracted normals, colors, and depth discontinuities. Note that a hand-drawn mask was used to isolate the object of interest.*

Acquisition with Hand-Held Flash: In order to demonstrate the practicality of obtaining RGBN images without specialized hardware, we use a setup consisting of a standard digital SLR camera and flash (Figure 2, left). In order to allow the flash to be moved by hand, we place mirror-reflective and white-diffuse spheres in the scene and use them to solve for flash position and intensity, respectively (a similar arrangement was used by Masselus et al. [2002]).

We usually capture 30–40 images with different flash positions, and a typical capture session takes only 2–3 minutes. The large number of images in a dataset provides considerable statistical redundancy, allowing for robust exclusion of shadows and specularities during the photometric stereo computation. In addition to extracting RGB albedo maps and per-pixel normals, we optionally use the method of Raskar et al. [2004] to find locations of depth discontinuities in the scene: these are locations of shadow boundaries *in the light direction*. A sample dataset is shown in Figure 2, right; note that this kind of object, with significant self-occlusion, would be difficult to capture using typical 3D range-scanners.

4 Tools for RGBN Processing

While specific rendering techniques are explored in the following section, many of these algorithms rely on two fundamental signal processing tools — smoothing and derivative estimation — as well as on the ability to effectively segment objects. Here we describe a toolbox of basic methods, analogous to those available in image processing and digital geometry processing, that are used by a variety of rendering algorithms. These tools (not the ones in Figure 16!) tend to offer the power and control of geometry-based methods, while retaining the simplicity and efficiency of image-based methods.

4.1 Filtering

Gaussian Filtering: A smoothing operator may be seen as the basic building block for many types of frequency-based methods, including denoising and scale-space analysis (i.e., multi-scale pyramids). The naive method for smoothing RGBN images would be to treat them as plain images with a 6D (color and normal) vector, in place of the conventional 3D (color only) vector at each pixel. One could then perform smoothing by convolving the image with a Gaussian (or, equivalently, simulating isotropic linear diffusion), and finally adjusting the normals to have unit length.

One problem with naive smoothing is due to foreshortening: regions with normals tilted away from the view direction will be smoothed more than they should be. With the formulation of smoothing as convolution, the naive method underestimates the "area" allocated to each pixel by $\cos\theta$, where θ is the angle between the normal and the view direction. Alternatively, in the linear diffusion formulation of smoothing, the distance between adjacent samples is underestimated by a factor of $\cos\theta$, hence the rate of diffusion is overestimated by this amount. In either case, correcting for foreshortening involves scaling the weight of each RGBN pixel, or decreasing the diffusion rate, by a factor of $\sec\theta$. This factor changes as smoothing progresses, leading to a nonlinear problem.

We therefore approximate the smoothing process with a linear one, by assuming that the view direction is constant across the image. For example, assume that the viewer is in the z direction. In this case, scaling the contribution of each normal by $\sec\theta$ transforms the vector (n_x, n_y, n_z) into $(n_x/n_z, n_y/n_z, 1)$. (In practice we use $n_z + \varepsilon$ in the denominator to avoid dividing by zero.) Performing convolution or isotropic diffusion on these vectors will leave the third component as 1, meaning that smoothing is now linear. This remapping also removes the need for explicit normalization at the end of the process, since there is a unique mapping from the first two components of the new vector to a unit-length vector in 3D.

Bilateral Filtering: While simple smoothing is sufficient for many applications, greater control and higher quality are often obtained with a filter that explicitly preserves edges. The bilateral filter [Tomasi and Manduchi 1998] is a non-iterative edge-preserving filter that bases the contribution of each pixel to the result on a *domain filter*, analogous to the Gaussian weight in standard smoothing, and a *range filter* that prevents the influence of pixels of significantly different intensities. The resultant color c'_i of pixel i is:

$$c'_i = \frac{\sum c_j \; g(|x_i - x_j|, \sigma_x)\; g(|c_i - c_j|, \sigma_c)}{\sum g(|x_i - x_j|, \sigma_x)\; g(|c_i - c_j|, \sigma_c)}, \qquad (3)$$

where c_i and x_i are the color and location of pixel i, g is a Gaussian, and the sum is over all pixels j in the image. In this equation, σ_x and σ_c are the widths of the domain and range filters, respectively; decreasing σ_c leads to better preservation of edges. The color differences may be computed in either a linear space such as RGB, or a perceptually-uniform space such as CIE-lab.

While we could apply a 6D bilateral filter to RGBN images directly, or apply separate filters to the colors and normals, we have observed situations in which it is profitable to apply a joint filter with separate control over color and normal similarity. For example, we may wish to avoid smoothing the colors across an edge that is visible only in the normal map (i.e., respecting discontinuities in shape rather than simply discontinuities in color). We achieve this effect by augmenting the bilateral filter with a term that reduces the influence of samples on the basis of differences in normals:

$$c'_i = \frac{\sum c_j \; g(|x_i - x_j|, \sigma_x)\; g(|c_i - c_j|, \sigma_c)\; g(|n_i - n_j|, \sigma_n)}{\sum g(|x_i - x_j|, \sigma_x)\; g(|c_i - c_j|, \sigma_c)\; g(|n_i - n_j|, \sigma_n)}. \qquad (4)$$

The normal differences $|n_i - n_j|$ are computed using the $1/n_z$ foreshortening correction, as above. An analogous equation is used to compute the filtered normal maps.

We use several variations of the bilateral filtering concept in our rendering pipeline, depending on the desired effect. For general-purpose smoothing of color and normals, the joint bilateral filter is applied. The traditional bilateral filter is used for enhancement in cases when it is important to smooth color only, such as to remove jagged color boundaries in our toon-shading implementation (Section 5.1). Figure 3 shows the difference between filtering on the basis of colors only, as compared to the joint filter.

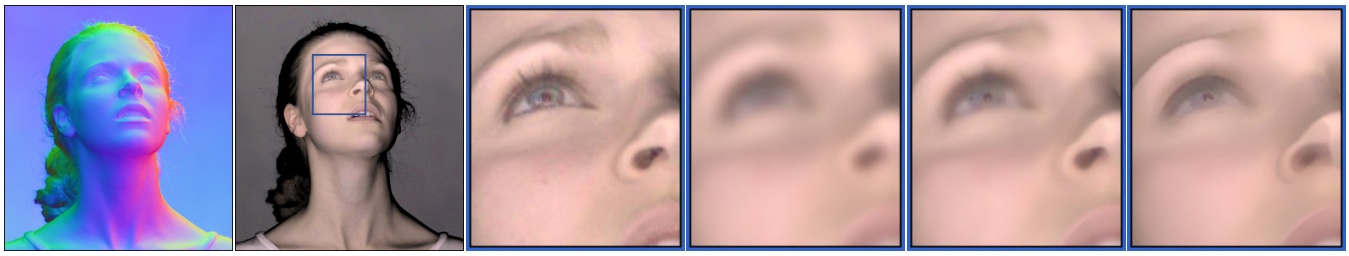


Figure 3: *The joint bilateral filter is capable of producing different results, depending on the settings of the domain and range filter widths. For large σ_c and σ_n, there is little edge preservation, and the filter resembles a simple Gaussian. Making σ_c small preserves color detail, such as that around the eye, while making σ_n small as well preserves both color and geometric edges.*

4.2 Curvature Estimation

Many algorithms for nonphotorealistic shading and line extraction make use of the curvatures of the surface and, in some cases, higher-order derivatives. As a brief review, let us recall that the *normal curvature* κ_n of a surface in some direction is the reciprocal of the radius of the circle that best approximates a normal slice of surface in that direction. The normal curvature varies with direction, but for a smooth surface it satisfies

$$\kappa_n = \frac{\begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} e & f \\ f & g \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix}}{\begin{pmatrix} s & t \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix}} = \frac{\begin{pmatrix} s & t \end{pmatrix} \mathbf{II} \begin{pmatrix} s \\ t \end{pmatrix}}{\begin{pmatrix} s & t \end{pmatrix} \mathbf{I} \begin{pmatrix} s \\ t \end{pmatrix}} \tag{5}$$

for any vector (s,t) expressed in terms of a tangent-plane coordinate system centered at the point. The symmetric matrices $\mathbf{I}$ and $\mathbf{II}$ appearing here, known as the first and second fundamental tensors, respectively, are therefore the basic quantities we wish to compute: knowing them, we may find the curvature in any given direction or compute other quantities such as mean curvature H (half the trace of $\mathbf{I}^{-1}\mathbf{II}$), Gaussian curvature K (the determinant), or the principal curvatures and principal directions.

The first fundamental tensor is defined as

$$\mathbf{I} = \begin{pmatrix} u \cdot u & v \cdot u \\ u \cdot v & v \cdot v \end{pmatrix}, \tag{6}$$

while the second fundamental tensor $\mathbf{II}$ is defined in terms of the directional derivatives of the surface normal:

$$\mathbf{II} = \begin{pmatrix} D_u n & D_v n \end{pmatrix} = \begin{pmatrix} \frac{\partial n}{\partial u} \cdot u & \frac{\partial n}{\partial v} \cdot u \\ \frac{\partial n}{\partial u} \cdot v & \frac{\partial n}{\partial v} \cdot v \end{pmatrix}, \tag{7}$$

where (u,v) are the axes of an arbitrary coordinate system in the tangent frame. This suggests that the first and second fundamental tensors may be estimated using finite differences, applying analogues of image edge detection kernels to the normal map. As with smoothing, however, we must be careful to account for the effect of foreshortening on the estimated curvatures. To do this, we take

$$u = \left(1, 0, -\tfrac{n_x}{n_z}\right), \quad v = \left(0, 1, -\tfrac{n_y}{n_z}\right) \tag{8}$$

in the above equations. These are vectors in the local tangent plane, and have the property that they project to the (unit-length) $\hat{x}$ and $\hat{y}$ directions in the image. Therefore, we may approximate the derivatives of the surface normal using finite differences:

$$\left(\tfrac{\partial n}{\partial u}\right)_{i,j} = \tfrac{1}{2}\left(n_{i+1,j} - n_{i-1,j}\right), \left(\tfrac{\partial n}{\partial v}\right)_{i,j} = \tfrac{1}{2}\left(n_{i,j+1} - n_{i,j-1}\right). \tag{9}$$

This formulation corresponds to the simplest (smallest-support) symmetric discrete derivative kernel; larger kernels (such as Sobel) may be used to provide smoother estimates.

4.3 Segmentation

Though segmentation is a fundamental part of image editing and compositing, we observe that segmenting images based purely on pixel intensities is inherently a hard problem: the same object may have dramatic variations in color, or lighting may cause two adjacent objects to not have a visible color discontinuity. With RGBN images, on the other hand, there is a second channel of information available: the normal map. For this reason, RGBN segmentation in fact typically produces better results than color-only segmentation.

While there are may classes of image segmentation algorithms available today, we choose to begin with the graph-partitioning algorithm described by Felzenszwalb and Huttenlocher [2004]. This algorithm is efficient ($O(n \log n)$ time), easily adaptable to incorporate dissimilarity functions based on both color and normal differences, and captures perceptually important regions while maintaining a global effect. Informally, this algorithm begins by building a graph corresponding to the image, with pixels at vertices and adjacent pixels connected by edges. Edge weights are assigned based on color dissimilarity, and a partitioning of the graph is sought. The algorithm partitions the graph such that between-segment dissimilarity (defined as the minimum edge weight between components) is greater than within-segment dissimilarity (defined as the largest weight of its minimum spanning tree, normalized by segment size and scaled by a user-selected constant).

In order to adapt this scheme for RGBN images, we modify the algorithm to include a dissimilarity function based on both normal and color differences, rather than on color only. In addition, we filter the image as a preprocess in order to remove artifacts due to noise. We tested several variations of the dissimilarity function and their effectiveness for segmenting the image. The greatest control was achieved when both color and normals were used as parameters to the blended dissimilarity function, and bilateral filtering was used for de-noising. The test case shown in Figure 4 was chosen to highlight the challenges of segmentation. A foreground object (rock) with mottled colors is placed over a patterned background object

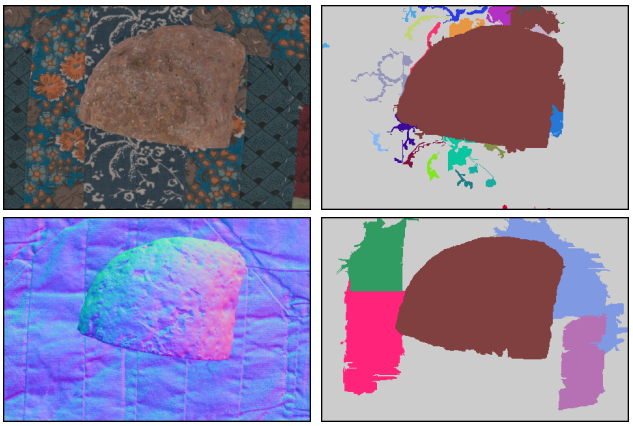


Figure 4: Top: *segmentation on the basis of color alone cannot separate the rock from the similarly-colored background.* ***Bottom:*** *using both normals and colors improves quality, permitting accurate segmentation.*

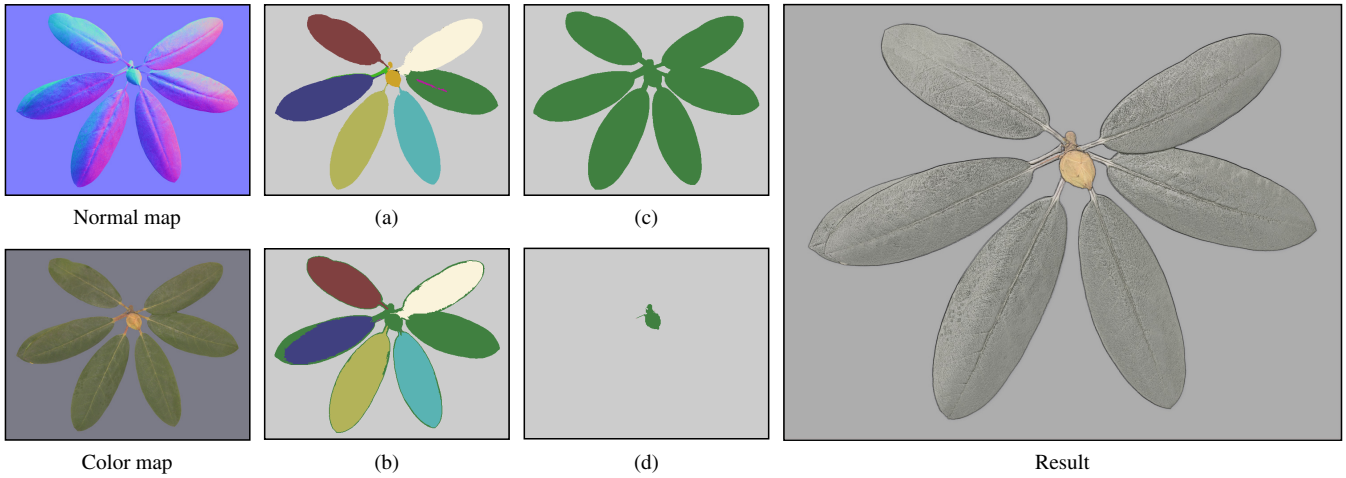


Figure 5: *A set of leaves with a bud is segmented on the basis of normals alone (a) or colors alone (b). The segmentation based on normals produces a cleaner result, while the color segmentation produces a sliver of incorrect segmentation around each leaf. Combined segmentation on the basis of normals and colors makes it possible to cleanly segment the object from the background (c) or the bud from the leaves (d), enabling stylization that varies for each segment.*

(quilt). Using color only we were unable to achieve an accurate segmentation between these objects; because one of the quilt colors is similar to shades in the rock, the rock segment tends to "spill" into these regions. On the other hand, adding normal dissimilarity into the technique produced the improved segmentation shown on the right. Figure 5 shows an application of segmentation to stylized rendering, in which the image was segmented completely automatically, then the user chose to apply a different style to one segment.

5 Stylized Depiction

Now that we have a signal processing framework for manipulating RGBN images, we can apply these tools for depiction.

5.1 Toon Shading

Cartoon shading, consisting of quantizing the amount of diffuse shading (i.e., $n \cdot l$) and mapping each discrete value to a different color, is a popular ingredient in nonphotorealistic renderings [Decaudin 1996]. While its roots lie in technical limitations of the cartoon and comics media, it remains popular because it abstracts shading while conveying information about geometry (the boundaries between toon shading regions are *isophotes* — curves of constant illumination — which have been shown to convey shape). Because toon shading is an effect that only depends on the surface normals, it easily extends to RGBN images (Figure 6). In practice, we use a bilateral filter to de-noise the normal maps, and further reduce the presence of jagged region boundaries by using a smoothstep function, instead of hard quantization.

Combining Toon Shading and Quantized Color: An important component of stylizing RGBN images is combining the shading from the shape with some version of the color data. There is little prior research that addresses the two complementary problems of:

- abstracting color in order to match a particular style of shading (e.g., toon shading), and
- combining color data with stylized shading generated from the normal map, so as to unambiguously convey the information present in both.

For toon shading we have investigated simply multiplying the shading calculation by the color (albedo) values, but this results in an un-natural image: it is difficult to interpret the toon shading in the presence of the smooth variations in color. Instead, we have found that it is beneficial to combine the toon shading with a *quantized* version of the color data, obtained by running a clustering algorithm. A sample result is shown in Figure 7: we have combined

Figure 6: ***Left:*** *toon shading computed from the normals in an RGBN image.* ***Right:*** *filtering of the normals and a softer step function lead to smoother results.*

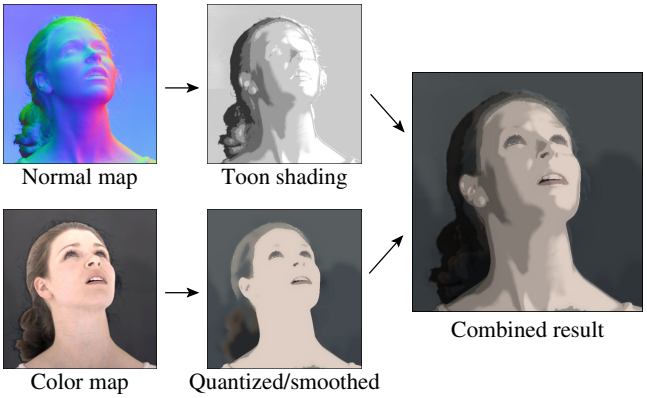


Figure 7: *Combining toon shading with quantized color.*

toon shading based on the normal map with a version of the color map that has been smoothed (using the color-only bilateral filter), then quantized to 16 colors. Figure 8 shows another example, with more smoothing applied to the color to produce a watercolor effect.

5.2 Line Drawings

A second class of rendering effects that we have investigated for RGBN images involves extraction of linear features. These are the "sparse, shape-conveying lines" referred to in Section 2, and include silhouettes, occluding contours, suggestive contours, and crest (ridge and valley) lines.

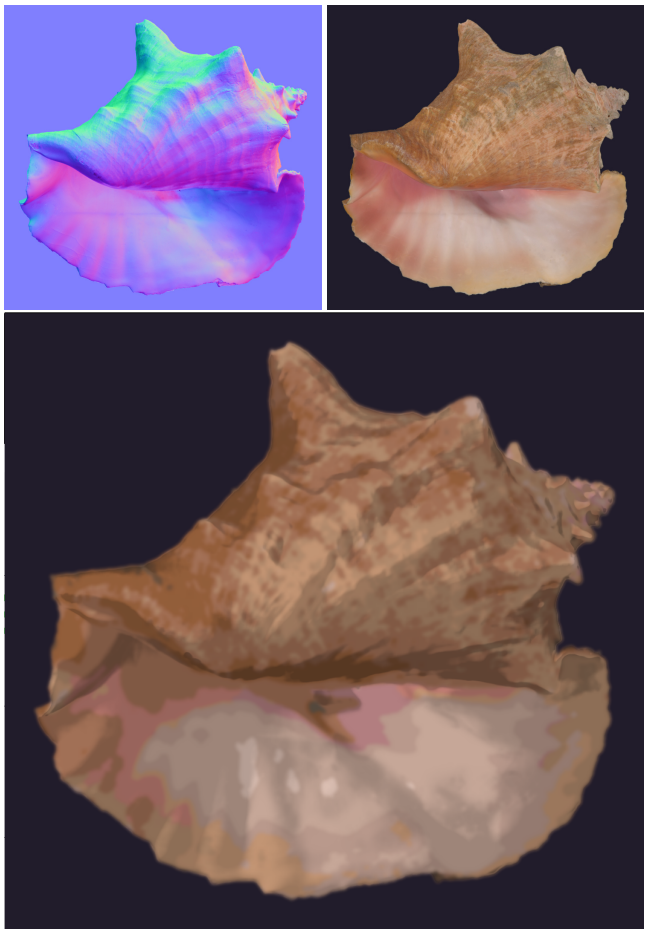


Figure 8: *Soft toon shading based on the normal map, combined with a smoothed version of the RGB color, yields a watercolor effect.*

Figure 9: *Locations of depth discontinuities overlaid on toon shading.*

Figure 10: *Rendering with suggestive contours. For relatively smooth objects (left), the object-space algorithm produces good results. For noisier input data, we begin with the results of the image-space algorithm (middle), then remove small connected components (right).*

Discontinuities: Silhouettes and occluding contours (i.e., locations of depth discontinuities) may be extracted either during the initial data analysis (as described in Section 3), or from the normal maps themselves. In cases in which extraction of discontinuities during acquisition is unreliable, we look for locations at which

- two adjacent pixels have very different normals, and
- one of those normals is nearly orthogonal to the view.

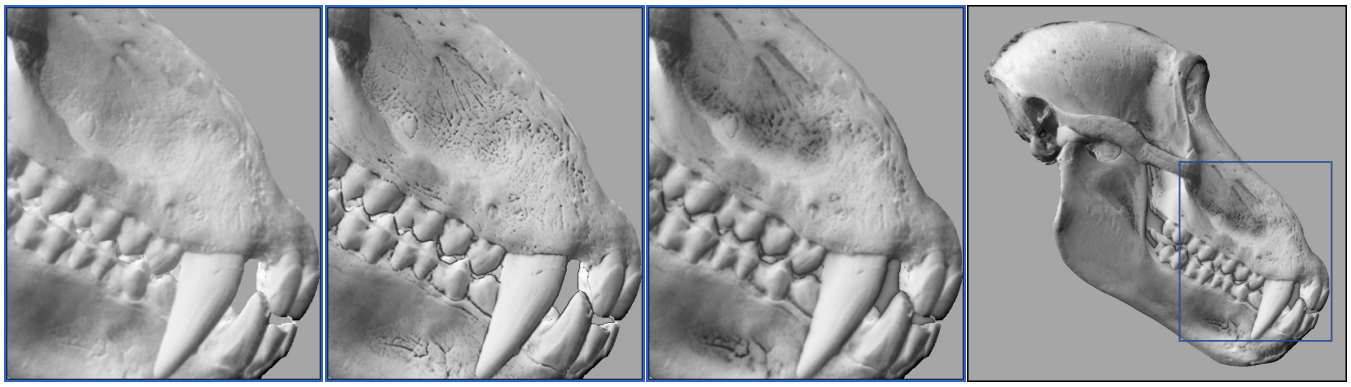
It is also possible to use color edges as additional cues to locate depth discontinuities, but we prefer not to rely on this: images in which there is relatively little color contrast are precisely those in which drawing lines adds significant visual information. Once the contours are extracted, they may be drawn with any line stylization, or may be overlaid on the results of any shading calculation (Figure 9).

Suggestive Contours: A second type of linear feature is the suggestive contour, introduced recently by DeCarlo et al. [2003; 2004]. Intuitively, these are "almost contours" — locations at which contours first appear with minimal change in viewpoint. Alternatively, they may be thought of as locations of intensity minima in a head-lit image (i.e., locations at which $n \cdot v$ is not zero, as it would be for contours, but is a local minimum). DeCarlo et al. describe two general algorithms for finding suggestive contours. Their object-space algorithm extracts zero-crossings of normal curvature in the projected view direction, whereas an image space version looks for intensity valleys in a head-lit image. We have explored both approaches for suggestive-contour extraction in RGBN images. Figure 10, left, shows the results of applying the object-space approach. We observe that this algorithm works well for smooth objects such as the dish, but is sensitive to noise in the data. Therefore, for noisy data sets such as the one shown in the middle, the image space algorithm offers more flexibility. After applying the joint bilateral filter we use a connected components pass to prioritize by length and prune short lines, giving the result shown on the right.

5.3 Curvature Shading and Shadows

Nonphotorealistic shading effects that convey cues about shadowing and indirect illumination are also frequently encountered, and include mean curvature shading [Kindlmann et al. 2003], accessibility shading [Miller 1994], ambient occlusion [Zhukov et al. 1998], and depth shading [Cohen et al. 2004]. While the details of these algorithms vary, broadly speaking they are all inspired by the intuitive observation that less light reaches valleys and folds on a surface. Hence, they darken indentations on the surface, and sometimes lighten bumps or ridges. Of these algorithms, the most practical to adapt to RGBN images is mean curvature shading, in which areas of negative mean curvature (concavities) are darkened, while areas of positive mean curvature are optionally lightened. Using the curvature-computation operators defined earlier, it is straightforward to perform mean curvature shading on RGBN images Figure 11.

One limitation of mean curvature shading as a method for conveying shape is that it only reveals high-frequency details. This is in contrast with methods such as ambient occlusion or accessibility shading, which are affected by features of many scales. Therefore,



Diffuse shading | Single-scale curvature shading | Multi-scale curvature shading | Multi-scale curvature shading

Figure 11: *Mean curvature shading based on the original curvatures (second image) reveals fine detail but does not convey a sense of overall shape. Multi-scale curvature shading (last two images) more closely resembles ambient occlusion, revealing shape over local neighborhoods.*

we have implemented *multi-scale* mean curvature shading, in which the curvature is computed at multiple levels of smoothing and the final color is an average of shading computed at different scales:

$$c_H = {}^1\!/_2 + {}^1\!/_2 \sum_{i=0}^{n} \underset{[-1\ldots c_{max}]}{\text{clamp}}\ a_i H_i, \tag{10}$$

where H_i is the mean curvature at the i-th scale, a_i are weights that may be equal across scales or chosen to emphasize high or low frequencies, and c_{max} is a parameter that may be set to 1 to both lighten bumps and darken concavities, or to 0 to only darken concavities. As shown in Figure 11, right, multi-scale curvature shading brings out the detail of differently-sized features simultaneously and compensates somewhat for the difficulty of implementing more global methods, such as ambient occlusion, for RGBN images.

Discontinuity Shadows: In order to provide a better sense of depth, we may also darken areas adjacent to the discontinuity lines computed during data acquisition. We simulate shadowing at these lines by using the fact that, as discussed by Raskar et al. [2004], we may infer not only the position but also the *direction* of the discontinuity. That is, we know on which side of the discontinuity line the occluding and occluded objects lie. Given this information, we may darken only the occluded side of the discontinuity line, achieving an effect similar to that demonstrated by the depth-buffer unsharp masking of Luft et al. [2006].

We begin with a *directional discontinuity map* $(d_x(x,y), d_y(x,y))$, in which the value at each pixel has magnitude proportional to the strength of the discontinuity and direction pointing towards the occluder. We compute our shadowing by convolution: the goal is to sum up kernel functions placed at each pixel, scaled and oriented according to (d_x, d_y). In order to compute this efficiently, we observe that, for any 1D function $\rho(r)$, the functions

$$\mathscr{S}_x(x,y) = x\rho(r), \quad \mathscr{S}_y(x,y) = y\rho(r), \quad \text{where } r = \sqrt{x^2+y^2}, \tag{11}$$

have the property that $(\mathscr{S}_x, \mathscr{S}_y) \cdot (d_x, d_y)$ is just a version of $\mathscr{S}_x$ rotated according to the given direction. We use

$$\rho(r) = e^{-r}/r \tag{12}$$

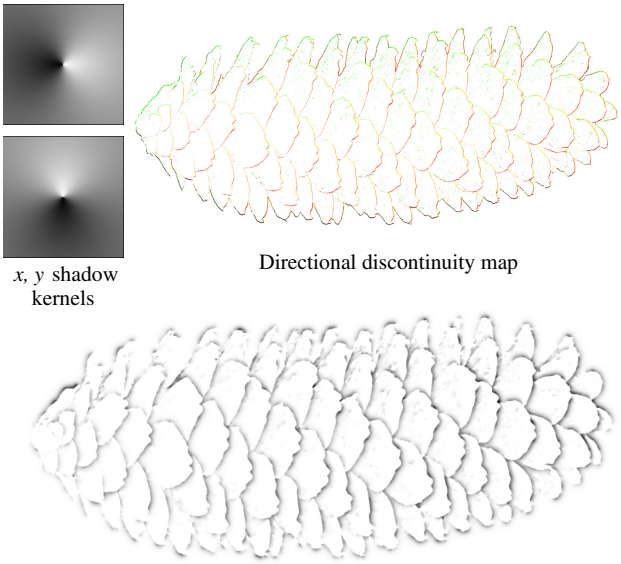
and compute

$$Shadow = 1 + \underset{[-1..0]}{\text{clamp}} \left[d_x * \mathscr{S}_x + d_y * \mathscr{S}_y \right]. \tag{13}$$

Figure 12 shows the shadow kernels, discontinuity map, and computed shadows for the pinecone dataset, while Figure 13 shows a result combining the shadows with exaggerated shading.

5.4 Exaggerated Shading

A final effect that we investigate is "exaggerated shading," which increases local contrast at all surface orientations and across all scales [Rusinkiewicz et al. 2006]. The computation uses several smoothed versions of surface normal maps, adjusting the effective light source position at each point based on the smoothed normals and principal directions. Using the smoothing and curvature-estimation operators described above, we achieve results such as those demonstrated in Figure 14.



x, y shadow kernels | Directional discontinuity map | Computed discontinuity shadows

Figure 12: *We convolve shadow kernels with our directional discontinuity map, then clamp to only negative values to yield our discontinuity shadows.*

6 Results and Discussion

We expect the techniques explored here to be applicable for easily creating illustrations of complex objects for which it is impractical to obtain full 3D scans, including domains such as historical documentation, botany, and medicine. For example, Figures 13 and 15 show botanical illustrations of the complex shapes of a pinecone and a leaf of chard, respectively. Similarly, Figure 16 shows an illustration of weathered tools in a style reminiscent of meticulous hand-shading, and revealing fine surface detail that is difficult to see in images and would be difficult to acquire in 3D scans. A further application is the study of archaeological objects. Figure 17 presents an analysis of a petroglyph, thought to be up to 9,000 years old, from the Legend Rock site in Wyoming. At right is a nonphotorealistic visualization of the fine relief of the surface. The compari-

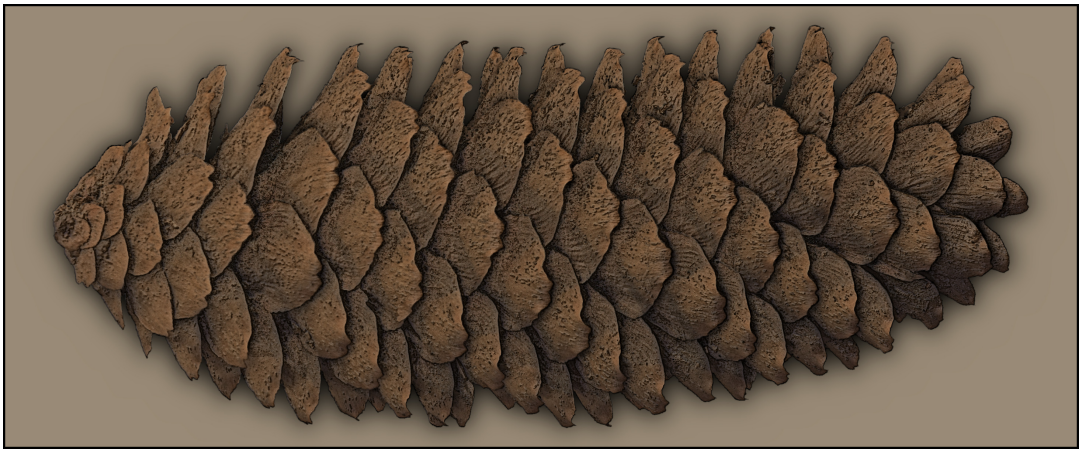


Figure 13: *Combination of exaggerated shading and discontinuity shadowing.*

son between it and the photograph at left reveals at a glance which inscriptions are fairly deep, and which are shallow, and reveals detail that is almost invisible in the color images.

Limitations: Several limitations of our technique are due to the acquisition process itself: it is difficult to acquire the surface normals of dark, shiny, translucent, or inter-reflecting objects, particularly when many different materials are simultaneously present in the scene and hence it is difficult to find heuristics for robust outlier rejection. The acquired normals may also be noisy, leading to difficulty in producing clean illustrations, especially for styles such as suggestive contour rendering. More fundamental to the RGBN datatype are limitations due to the lack of depth information: it is difficult or impossible to change the view, or to compute cast shadows except for very local ones. Finally, the RGBN images considered here only store a single (diffuse) color per pixel, though one may imagine extensions to support specularity (RGB^2N images) or arbitrary BRDFs per pixel. Despite these drawbacks, we believe that the variety of datasets and styles examined in this paper demonstrates the wide applicability of RGBN acquisition and stylized rendering.

Figure 14: *Exaggerated shading, with warm-to-cool color transition for the low frequencies.*

7 Conclusion and Future Work

This paper describes the RGBN image data type combining albedo and normals. RGBN images, which lie somewhere between 2D and full 3D, offer more powerful rendering capabilities than regular RGB images, coupled with ease of acquisition not attributed to full 3D models. We show how to adapt existing image processing algorithms to filter RGBN data effectively, and describe how these algorithms form the basis for a toolbox of stylization effects. We apply these effects in combination to produce a variety of illustrations for a range of acquired objects.

This work suggests an number of areas for future research:

Improved Stylization: RGBN images should be promoted to first-class citizens in conventional image editing programs in order to facilitate access to the combinatorial strength of the variety of algorithms described. Also, as mentioned in Section 5.2, suggestive contours are sensitive to noisy data. Therefore, for acquired RGBN images it would be helpful to devise new algorithms for greater stability in extracted suggestive contours. One possible strategy is to look for such lines that persist across scales, in a fashion similar to that used by Jeong et al. [2005] for LOD control of line drawings.

RGBN/t: The video analog of RGB also presents a unique set of challenges and opportunities. The acquisition side is difficult because we have to assume that the scene is in motion. On the rendering side await a host of stylization effects in the spirit of the work of Wang et al. [2004].

Operations on RGBN Images: Researchers have made great strides in the analysis and synthesis of texture in RGB images (e.g. [Efros and Leung 1999]). Of course such algorithms should apply directly in the RGBN domain. However, perhaps more interesting would be the use of a known RGB channel to hallucinate or inpaint areas of missing normals, or vice versa.

References

BARTESAGHI, A., SAPIRO, G., MALZBENDER, T., AND GELB, D. 2005. Three-Dimensional Shape Rendering from Multiple Images. *Graphical Models, Vol. 67, No. 4* (July).

BASRI, R., AND JACOBS, D. 2001. Photometric Stereo with General, Unknown Lighting. In *Proc. CVPR*.

BERNARDINI, F., MARTIN, I., MITTLEMAN, J., RUSHMEIER, H., AND TAUBIN, G. 2002. Building a Digital Model of Michelangelo's Florentine Pietá. *IEEE Computer Graphics and Applications, Vol. 22, No. 1*, 59–67.

COHEN, J., DUNCAN, D., SNYDER, D., COOPER, J., KUMAR, S., HAHN, D., CHEN, Y., PURNOMO, B., AND GRAETTINGER, J. 2004. iClay: Digitizing Cuneiform. In *Proc. Symposium on Virtual Reality, Archaeology, and Cultural Heritage (VAST)*.

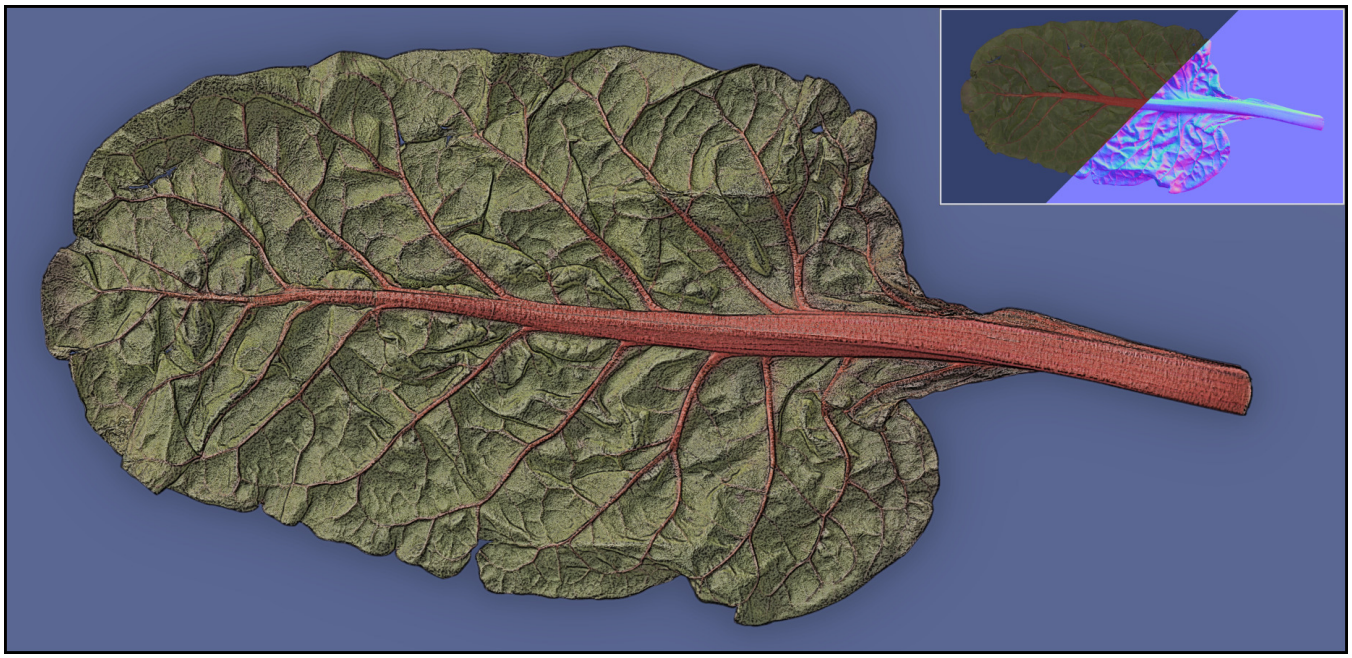


Figure 15: *A leaf of rainbow chard, stylized using a combination of exaggerated shading and suggestive contour lines.*

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. 2000. Acquiring the Reflectance Field of a Human Face. In *Proc. ACM SIGGRAPH*.

DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive Contours for Conveying Shape. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 22, No. 3*, 848–855.

DECARLO, D., FINKELSTEIN, A., AND RUSINKIEWICZ, S. 2004. Interactive Rendering of Suggestive Contours with Temporal Coherence. In *Proc. NPAR*.

DECAUDIN, P. 1996. Cartoon-Looking Redering of 3D-Scenes. Tech. Rep. 2919, INRIA.

DEERING, M., WINNER, S., SCHEDIWY, B., DUFFY, C., AND HUNT, N. 1988. The Triangle Processor and Normal Vector Shader: A VLSI System for High Performance Graphics. In *Proc. SIGGRAPH*.

DOOLEY, D., AND COHEN, M. F. 1990. Automatic Illustration of 3D Geometric Models: Lines. In *Proc. I3D*.

EFROS, A., AND LEUNG, T. 1999. Texture Synthesis by Non-Parametric Sampling. In *Proc. ICCV*.

ELBER, G., AND COHEN, E. 1990. Hidden Curve Removal for Free Form Surfaces. In *Proc. SIGGRAPH*.

FELZENSZWALB, P., AND HUTTENLOCHER, D. 2004. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision, Vol. 59, No. 2* (Sept.).

GEORGHIADES, A. 2003. Incorporating the Torrance and Sparrow Model of Reflectance in Uncalibrated Photometric Stereo. In *Proc. ICCV*.

GOOCH, A., GOOCH, B., SHIRLEY, P., AND COHEN, E. 1998. A Non-Photorealistic Lighting Model for Automatic Techical Illustration. In *Proc. ACM SIGGRAPH*.

HERTZMANN, A., AND ZORIN, D. 2000. Illustrating Smooth Surfaces. In *Proc. ACM SIGGRAPH*, 517–526.

HORN, B. K. P. 1970. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, Massachusetts Institute of Technology.

HORN, B. K. P. 1981. Hill Shading and the Reflectance Map. *Proc. IEEE, Vol. 69, No. 1*, 14–47.

INTERRANTE, V., FUCHS, H., AND PIZER, S. 1995. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. In *Proc. IEEE Visualization*.

JEONG, K., NI, A., LEE, S., AND MARKOSIAN, L. 2005. Detail Control in Line Drawings of 3D Meshes. *The Visual Computer, Vol. 21, No. 8–10*, 698–706.

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*.

LEE, C. H., HAO, X., AND VARSHNEY, A. 2006. Geometry-Dependent Lighting. *IEEE Trans. Visualization and Computer Graphics, Vol. 12, No. 2*.

LUFT, T., COLDITZ, C., AND DEUSSEN, O. 2006. Image Enhancement by Unsharp Masking the Depth Buffer. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 25, No. 3*.

MALZBENDER, T., GELB, D., AND WOLTERS, H. 2001. Polynomial Texture Maps. In *Proc. ACM SIGGRAPH*.

MALZBENDER, T., WILBURN, B., GELB, D., AND AMBRISCO, B. 2006. Surface Enhancement Using Real-time Photometric Stereo and Reflectance Transformation. In *Proc. Eurographics Symposium on Rendering*.

MARKOSIAN, L., KOWALSKI, M., TRYCHIN, S., BOURDEV, L., GOLDSTEIN, D., AND HUGHES, J. 1997. Real-Time Nonphotorealistic Rendering. In *Proc. ACM SIGGRAPH*, 415–420.

MASSELUS, V., DUTRÉ, P., AND ANRYS, F. 2002. The Free-Form Light Stage. In *Proc. Eurographics Rendering Workshop*.

MILLER, G. 1994. Efficient Algorithms for Local and Global Accessibility Shading. In *Proc. SIGGRAPH*.

MUDGE, M., VOUTAZ, J.-P., SCHROER, C., AND LUM, M. 2005. Reflection Transformation Imaging and Virtual Representations of Coins from the Hospice of the Grand St. Bernard. In *Proc. Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*.

MUDGE, M., MALZBENDER, T., SCHROER, C., AND LUM, M. 2006. New Reflection Transformation Imaging Methods for Rock Art and Multiple-Viewpoint Display. In *Proc. Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, H.-P. 2004. Ridge-Valley Lines on Meshes via Implicit Surface Fitting. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 23, No. 3*.

PAULY, M., KEISER, R., AND GROSS, M. 2003. Multi-Scale Feature Extraction on Point-Sampled Models. In *Proc. Eurographics*.

RASKAR, R., TAN, K.-H., FERIS, R., YU, J., AND TURK, M. 2004. Non-Photorealistic Camera: Depth Edge Detection and Stylized Rendering using Multi-Flash Imaging. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 23, No. 3*.

RHEINGANS, P., AND EBERT, D. 2001. Volume Illustration: Nonphotorealistic Rendering of Volume Models. *IEEE Trans. Vis. Comp. Gr., Vol. 7, No. 3*, 253–264.

RUSHMEIER, H., AND BERNARDINI, F. 1999. Computing Consistent Normals and Colors from Photometric Data. In *Proc. 3DIM*.

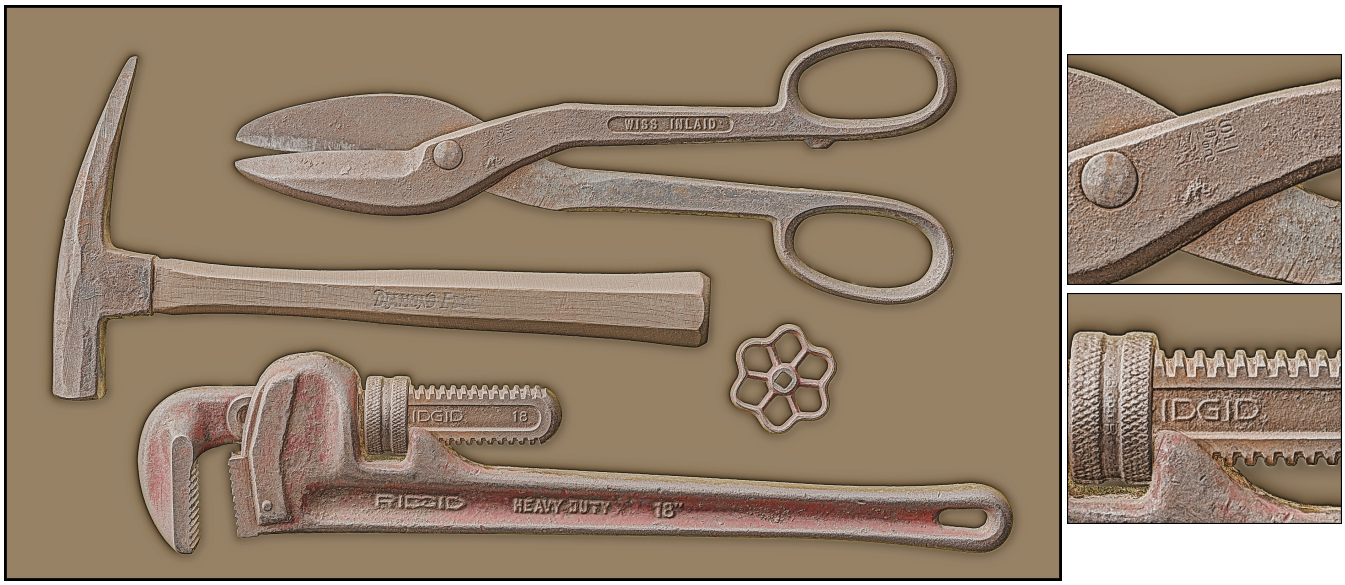


Figure 16: *Illustration of tools reveals fine details, such as the maker's stamp on the shears.*

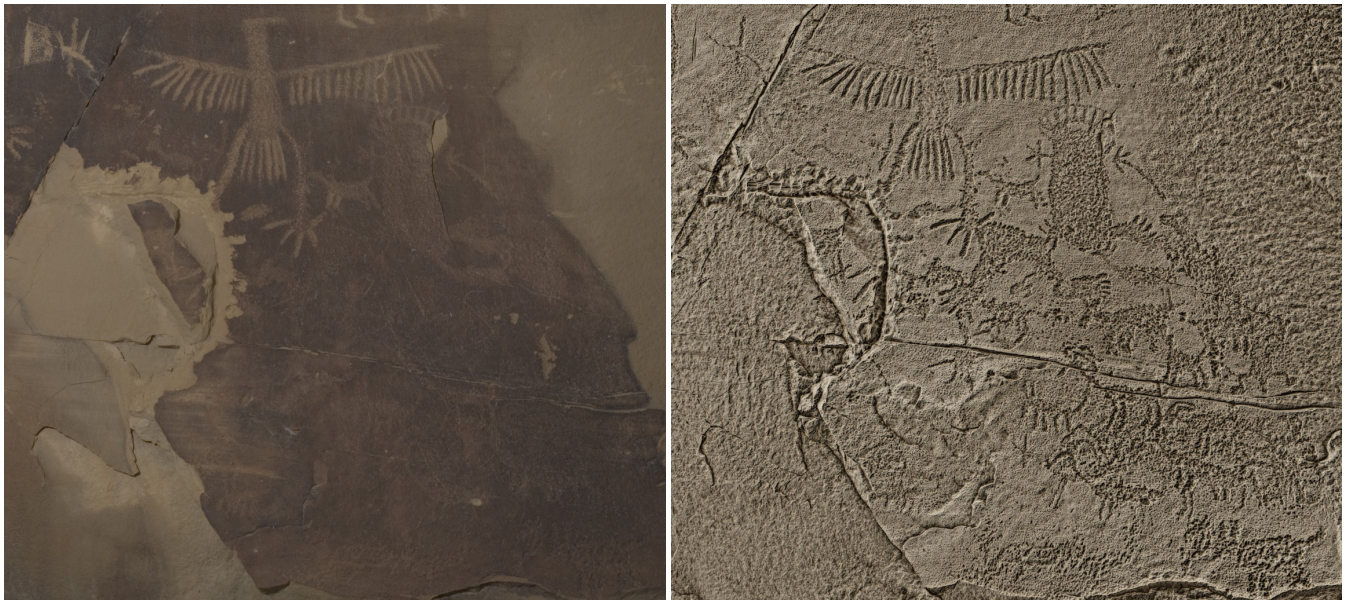


Figure 17: *Color image and nonphotorealistic rendering (with mean curvature shading and exaggerated shading) of the Legend Rock archaeological site.*

RUSINKIEWICZ, S., BURNS, M., AND DECARLO, D. 2006. Exaggerated Shading for Depicting Shape and Detail. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 25, No. 3*.

SAITO, T., AND TAKAHASHI, T. 1990. Comprehensible Rendering of 3-D Shapes. In *Proc. ACM SIGGRAPH*, 197–206.

SLOAN, P.-P., MARTIN, W., GOOCH, A., AND GOOCH, B. 2001. The Lit Sphere: A Model for Capturing NPR Shading from Art. In *Proc. Graphics Interface*.

TAGARE, H., AND DEFIGUEIREDO, R. J. P. 1991. A Theory of Photometric Stereo for a Class of Diffuse Non-Lambertian Surfaces. *IEEE Trans. PAMI, Vol. 13, No. 2*, 133–152.

TAN, K.-H., KOBLER, J., FERIS, R., DIETZ, P., AND RASKAR, R. 2004. Shape-Enhanced Surgical Visualizations and Medical Illustrations with Multi-Flash Imaging. In *Proc. MICCAI*.

THIRION, J.-P., AND GOURDON, A. 1996. The 3D Marching Lines Algorithm. *Graphical Models and Image Processing, Vol. 58, No. 6* (Nov.), 503–509.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral Filtering for Gray and Color Images. In *Proc. ICCV*.

WANG, J., XU, Y., SHUM, H.-Y., AND COHEN, M. F. 2004. Video Tooning. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 23, No. 3*, 574–583.

WENGER, A., GARDNER, A., TCHOU, C., UNGER, J., HAWKINS, T., AND DEBEVEC, P. 2005. Performace Relighting and Reflectance Transformation with Time-Multiplexed Illumination. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 24, No. 3*.

WEYRICH, T., MATUSIK, W., PFISTER, H., BICKEL, B., DONNER, C., TU, C., MCANDLESS, J., LEE, J., NGAN, A., JENSEN, H. W., AND GROSS, M. 2006. Analysis of Human Faces using a Measurement-Based Skin Reflectance Model. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 25, No. 3*.

WINKENBACH, G., AND SALESIN, D. 1994. Computer-Generated Pen-and-Ink Illustration. In *Proc. SIGGRAPH*.

WOODHAM, R. 1980. Photometric Method for Determining Surface Orientation from Multiple Images. *Optical Engineering, Vol. 19, No. 1*, 139–144.

ZHUKOV, S., IONES, A., AND KRONIN, G. 1998. An Ambient Light Illumination Model. In *Proc. Eurographics Rendering Workshop*.