Illustration of Complex Real-World Objects using Images with Normals

Corey Toler-Franklin Adam Finkelstein Szymon Rusinkiewicz
Princeton University

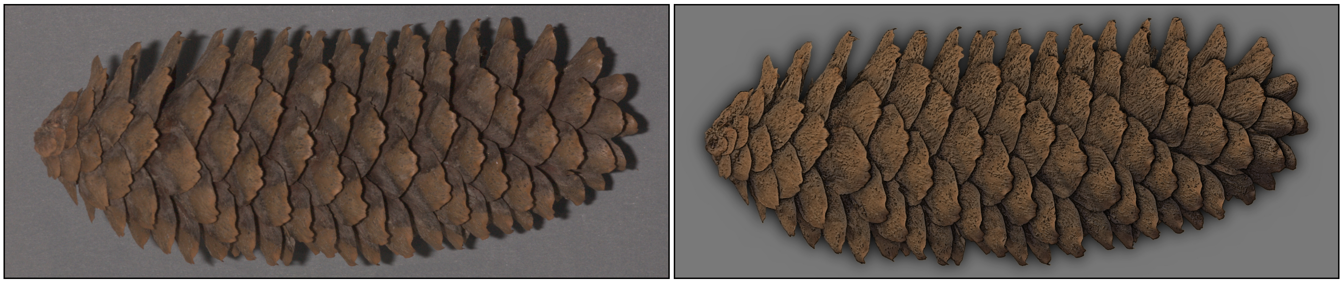


Figure 1: *Left: a photograph of a pinecone. Right: an illustration utilizing exaggerated shading and discontinuity shadows. The latter reveals both subtle details and depth complexity more clearly, as is common in hand-drawn scientific illustrations. Our pipeline, based on acquired images with per-pixel normals, produces such illustrations easily and with high fidelity; it would be difficult to generate equivalent results starting with either a single 2D image or a 3D scan.*

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 also introduce new stylization effects based on multiscale mean curvature shading, as well as fast discontinuity shadows. We show that our rendering pipeline can produce detailed yet understandable illustrations in medical, technical, and archaeological domains.

1 Introduction

Scientific illustrators use a variety of non-photorealistic techniques, including stylized shading and line drawing, to produce detailed yet comprehensible images [Hodges 2003]. Such illustrations are common in domains such as documenting archaeological excavations, or in textbooks on medicine or botany. However, consider how one might produce an illustration of the pinecone shown in Figure 1, right. In contrast to the photograph at left, this illustration reveals detail by emphasizing the small ridges present on each plate (visible when zooming in on the electronic version of this paper), and improves overall comprehensibility by introducing consistent soft shadows. While it would be possible to produce this type of illustration by hand, it would require many hours of meticulous drawing and shading. The illustration could not be produced automatically from a photograph, because the effects require more information than is present in a 2D color image. Existing non-photorealistic rendering (NPR) methods would require starting with a 3D model, but capturing a model of the pinecone using a 3D scanner is difficult. This is both because of the significant occlusion present in the pinecone (which would require combining dozens of scans to obtain a hole-free model, even from one viewpoint) and because existing scanners have neither the resolution nor the low noise necessary to capture subtle details such as the ridges (the resolution of the images in Figure 1 is approximately 20 pixels per millimeter).

We argue that the most practical method for producing illustrations such as those shown in Figure 1, right, is to acquire a type of data lying between simple 2D images and full 3D models: images with a color albedo and a surface normal stored at each pixel ("RGBN images"). Such datasets may be captured using *photometric stereo* [Woodham 1980], in which normals are inferred from several images (captured from a single camera position) of an object illuminated from different directions. Moreover, we demonstrate how to derive signal processing (smoothing and derivative estimation) and segmentation algorithms for RGBN images, enabling a broad spectrum of stylized rendering methods. In short, RGBN images 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).

RGBN 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 captured images with normals at sufficiently high quality to be used for cinematic relighting. In this paper we build upon previous work on *realistic* relighting to investigate the generation of *stylized* renderings from RGBN images. Our contributions include:

- tools for segmentation and signal processing on RGBN images, including smoothing and derivative estimation (Section 4);
- adaptation of (existing) shading and line drawing styles to RGBN images (Section 5);
- new non-photorealistic rendering styles motivated by the limitations of RGBN images, namely multiscale mean curvature shading and fast discontinuity shadows (Section 5.2);
- demonstration of our pipeline for producing detailed yet understandable illustrations (Section 6).

While much of the research in NPR has focused on elision of extraneous detail, this project also addresses the ability to enhance fine detail in an illustration. Thus several of the figures (1,10,12-14) reveal texture and structure at a fine scale largely unseen in previous work in NPR, yet critical for illustrations in, for example, archeology or botany whose goals are to communicate such detail. We believe that it would be difficult to achieve such effects starting from either traditional photographs or acquired 3D models.

2 Previous Work

Non-Photorealistic Rendering of 3D Models: 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].

A 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) [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 [DeCarlo et al. 2003; DeCarlo et al. 2004], which complement contours to better depict shape.

All of the above NPR algorithms have been investigated only for 3D models (and in some cases for volumetric data, such as [Interrante et al. 1995; Ebert and Rheingans 2000; Kindlmann et al. 2003]). Therefore, because of the difficulty of 3D modeling and acquisition, they have rarely been applied for the task of illustrating real-world objects. In this paper, we derive the signal processing tools necessary to adapt many of these methods for RGBN images, and demonstrate the algorithms on complex real-world objects.

Depiction of Images with Discontinuities: One class of NPR algorithms that operate on more than simply colors includes techniques that 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 develop a fast discontinuity shadowing effect in this paper, using contours 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 to obtain positions, this is typically unstable without the additional availability of coarse geometry to constrain the reconstruction [Nehab et al. 2005]. Therefore, several applications have 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], or by fitting a polynomial to measurements [Malzbender et al. 2001]. 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. We build upon these algorithms to develop a suite of geometric processing tools for RGBN images, then applying them to a variety of NPR styles.

3 Acquisition

While the RGBN rendering tools we develop may use datasets obtained using any technique, we use photometric stereo data. We briefly describe the technique, as well as our acquisition apparatus.

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. 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 l_i, yielding intensities e_i. Using the Lambertian lighting law

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

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

$$\begin{pmatrix} l_{1,x} & l_{1,y} & l_{1,z} \\ l_{2,x} & l_{2,y} & l_{2,z} \\ l_{3,x} & l_{3,y} & l_{3,z} \\ & \vdots & \end{pmatrix} \begin{pmatrix} a\,\hat{n}_x \\ a\,\hat{n}_y \\ a\,\hat{n}_z \end{pmatrix} = \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \end{pmatrix}. \tag{2}$$

This equation may be solved using least squares for the vector $a\hat{n}$, whose length and direction give a and $\hat{n}$, respectively. Typically, more than the minimum three lights are used, providing noise reduction and allowing for outlier (specularity and shadow) rejection.

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 specular-

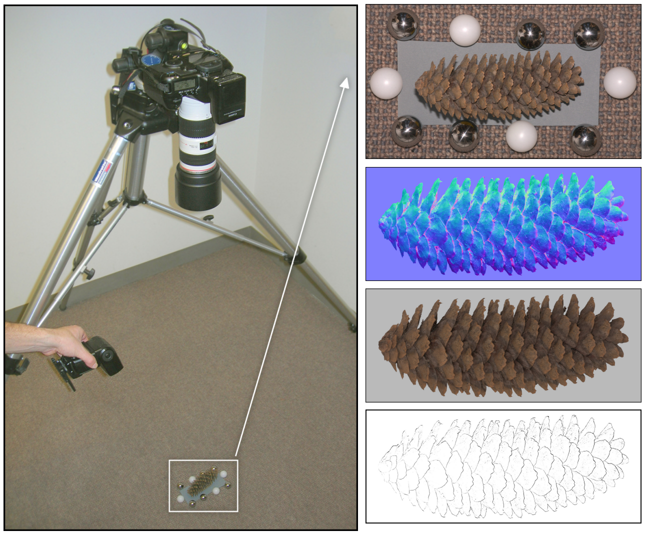
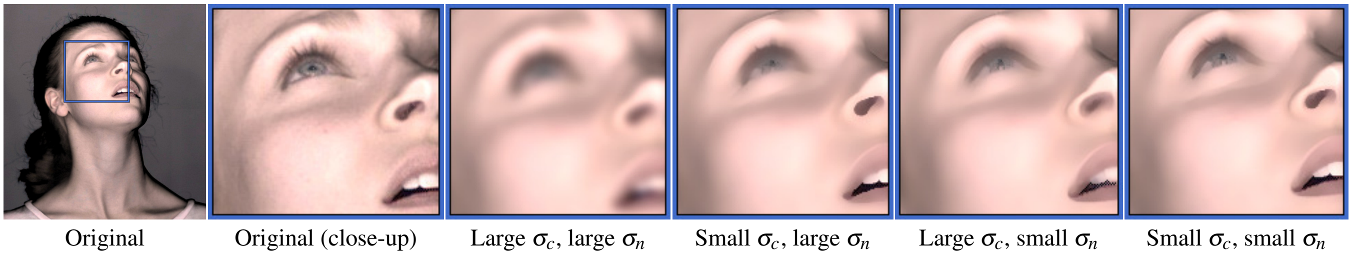


***Figure 2: Left:** our setup for capturing RGBN images, using a digital SLR camera and hand-held flash. White and mirror spheres are used to find the flash intensity and position for each captured image. **Right:** An original image, together with extracted normals, colors, and depth discontinuities.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



Original Original (close-up) Large σ_c, large σ_n Small σ_c, large σ_n Large σ_c, small σ_n Small σ_c, small σ_n

Figure 3: *The RGBN 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.*

ities during the photometric stereo computation (i.e., equation 2). We are thus able to recover normals even for highly-occluded geometry for which only a small percentage of light positions are visible from each pixel (e.g., the pinecone in Figure 1) and materials with a significant specular component (e.g., the helmet in Figure 7). In addition to extracting RGB albedo maps and per-pixel normals, we 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*. The results are shown in Figure 2, right. Note that a hand-drawn mask was used to isolate the object of interest.

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. Therefore, we begin by describing a toolbox of methods analogous to those available in image processing and digital geometry processing. These tools 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 is a basic building block for many 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 vector at each pixel. One could then perform smoothing by convolving the image with a Gaussian 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:

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

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:

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

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.

By varying the bilateral filter parameters, we obtain a variety of effects appropriate for different rendering styles. Figure 3 compares several parameter settings: note that using a small σ_c preserves sharp color boundaries, such as the pupil, while additionally making σ_n small emphasizes creases, such as the eyebrow.

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, are therefore the basic quantities we wish to compute: knowing them, we may find the curvature in any direction or compute quantities such as mean curvature (half the trace of $\mathbf{I}^{-1}\mathbf{II}$), Gaussian curvature (the determinant), or the principal curvatures and 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:

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

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 weighted by color dissimilarity. 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 (using the bilateral filter) in order to remove artifacts due to noise. Incorporating these changes, we are able to produce accurate segmentations even when there are no visible edges in color. For example, in Figure 4 we are able to segment the hammer and shears into meaningful facets despite the lack of color information. Figure 5 shows an application of segmentation to stylized rendering, in which the image was segmented automatically, then the user chose to apply a different style to one segment.

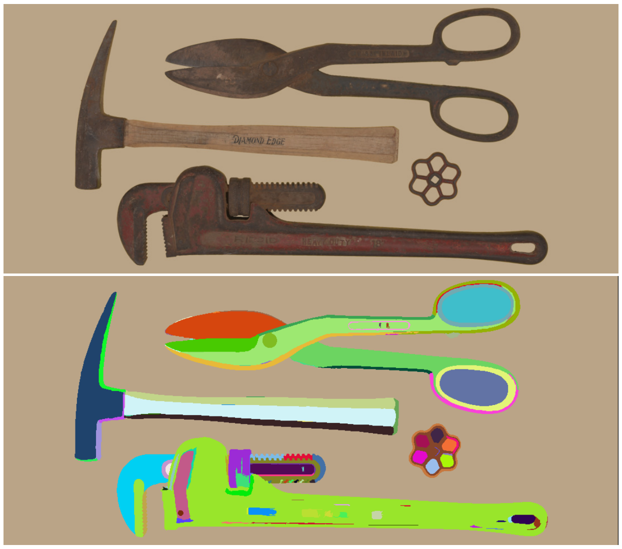


Figure 4: *RGBN segmentation can produce accurate results even when there are no visible color edges. For example, in the segmentation result at bottom, note that the hammer has been segmented into multiple facets.*

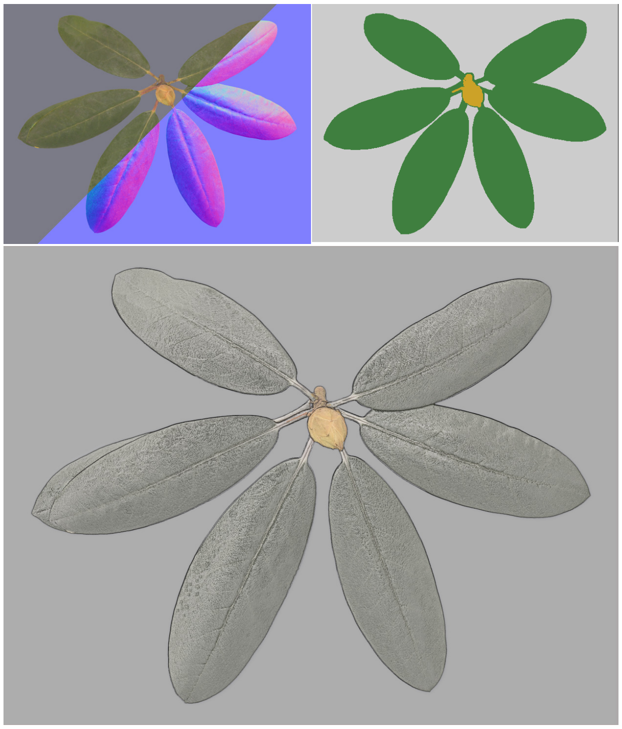


Figure 5: *A dataset of leaves with a bud (top left) is segmented on the basis of both colors and normals (top right) to separate the bud from the leaves. This enables applying a different stylization for each segment (bottom).*

5 Stylized Depiction

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

5.1 Adaptations of Existing Algorithms

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 be-

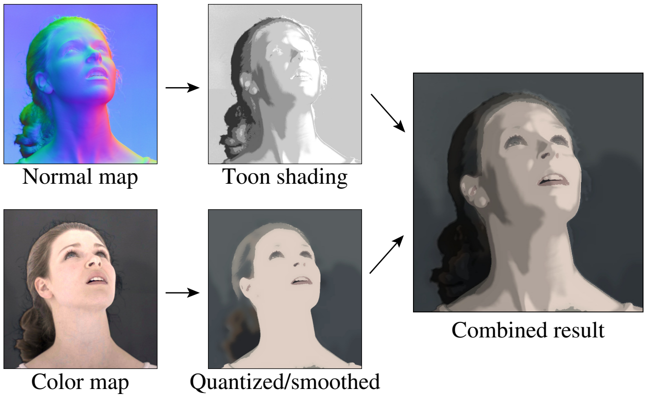


Figure 6: *Combining toon shading with quantized color.*

cause 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. We use a bilateral filter to de-noise the normal maps, and further reduce the presence of jagged region boundaries by using a smooth-step function, instead of hard quantization.

In order to incorporate the color (albedo) information into the toon shading, we have investigated simply multiplying the shading calculation by the color (albedo) values, but this results in an unnatural 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. One result is shown in Figure 6: we have combined 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.

Discontinuity Lines: 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 (their dot product is below a threshold), and
- one of those normals is nearly orthogonal to the view (its dot product with the view is within a threshold of zero).

Figure 7 shows discontinuity lines together with toon shading: note that this highlights *geometric* discontinuities as opposed to *color* discontinuities.

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 8 shows the results of applying the image-space algorithm. We observe that this algorithm works well for bumpy surfaces, since it allows for flexibility in cleaning up short lines (the figure uses bilateral filtering and a connected-components pass to prioritize by length and prune short lines). For smoother objects, such as the dish in Figure 9, the

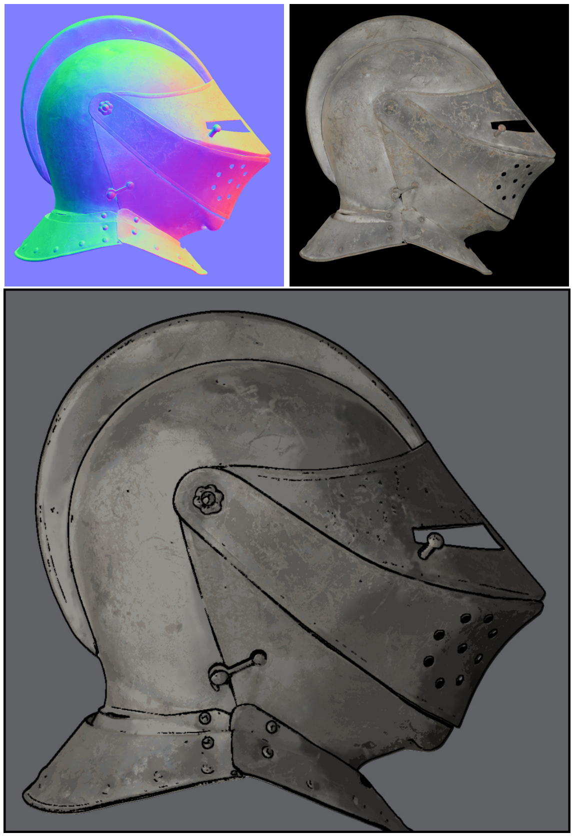


Figure 7: *Locations of depth discontinuities overlaid on toon shading.*

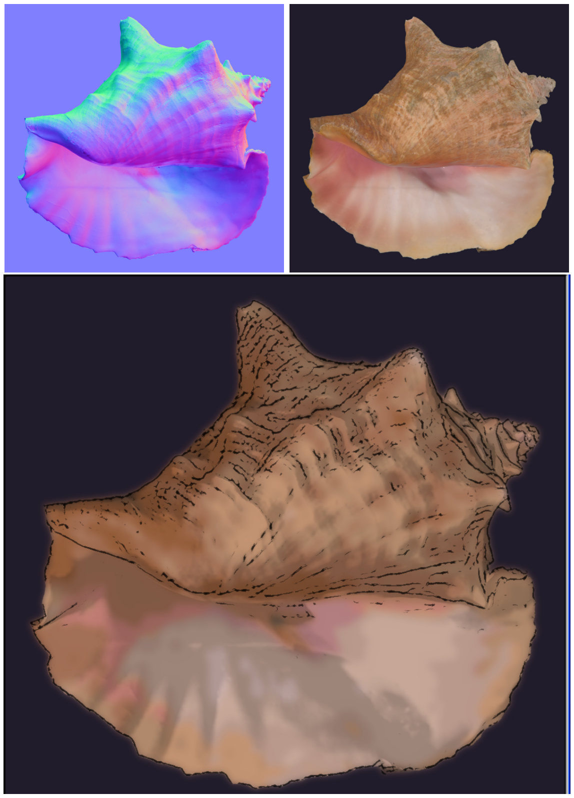


Figure 8: *Soft toon shading based on the normal map, combined with a smoothed version of the RGB color, yields a watercolor effect. Suggestive contour lines are added to emphasize features.*

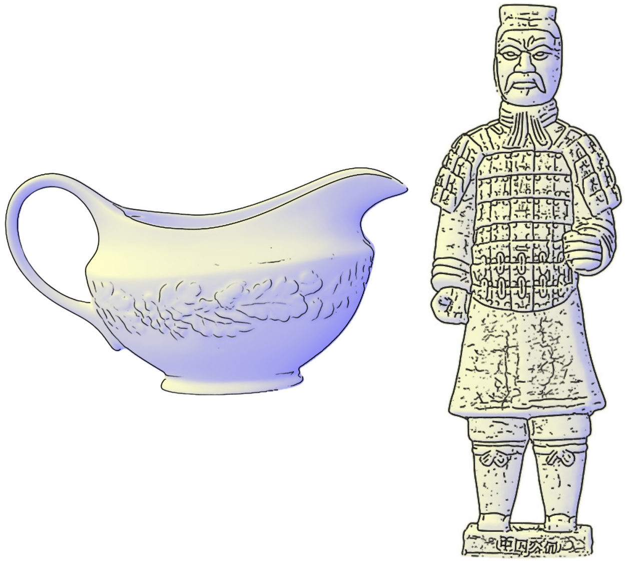


Figure 9: *Renderings with suggestive contours (left) and valley lines (right).*

object-space algorithm (using the curvatures and curvature derivatives computed above) yields good results.

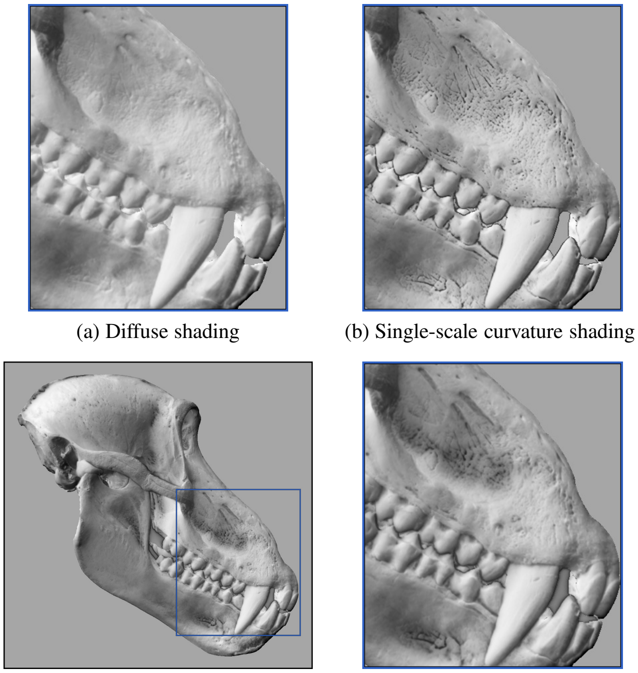
Ridge and Valley Lines: We also extract crest lines (ridges and valleys), which are defined as local maxima of the greatest principal curvature, in the corresponding principal direction. This is implemented by finding locations at which one of the components of the derivative-of-curvature tensor is zero, together with additional conditions on principal curvature [Ohtake et al. 2004]. A sample result is shown in Figure 9, right.

Exaggerated Shading: A final effect that we adapt to RGBN images is "exaggerated shading," which increases local contrast at all surface orientations and across all scales [Rusinkiewicz et al. 2006]. The computation uses several smoothed 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 Figures 1, 12, 13, and 14.

5.2 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. Many of these algorithms, however, require knowing depth and hence are impractical to adapt to RGBN images. Therefore, we introduce two new algorithms that produce similar effects, yet may be used with RGBN images.

Multiscale Curvature Shading: Of the above 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 (Figure 10b). However, 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, we introduce *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:



(a) Diffuse shading

(b) Single-scale curvature shading

(c) Multi-scale curvature shading

(d) Multi-scale curvature shading

Figure 10: *Mean curvature shading based on the original curvatures (b) reveals fine detail but does not convey a sense of overall shape. Multi-scale curvature shading (c and d) more closely resembles ambient occlusion, revealing shape over local neighborhoods.*

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

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 10 c and d, 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.

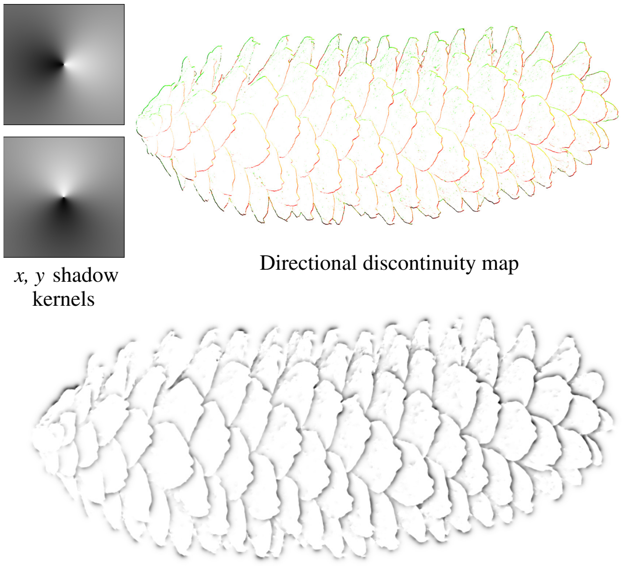
Fast Discontinuity Shadows: In order to provide a better sense of depth, we may also darken areas adjacent to the discontinuity lines computed during data acquisition. We simulate shadowing at these lines by using the fact that, as discussed by Raskar et al. [2004], we may infer not only the position 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* $\big(d_x(x,y),\, d_y(x,y)\big)$, 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

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

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

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



x, y shadow kernels

Directional discontinuity map

Computed discontinuity shadows

Figure 11: *We convolve shadow kernels with our directional discontinuity map, then clamp to only negative values to yield our discontinuity shadows.*

and compute

$$Shadow = 1 + \operatorname*{clamp}_{[-1..0]} \big[d_x * \mathscr{S}_x + d_y * \mathscr{S}_y) \big]. \qquad (13)$$

Figure 11 shows the shadow kernels, discontinuity map, and computed shadows for the pinecone dataset, while Figure 1 shows a result combining the shadows with exaggerated shading.

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 1 and 12 show botanical illustrations of the complex shapes of a pinecone and a leaf of chard, respectively. Similarly, Figure 13 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 14 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 comparison 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 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.

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 above, 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).

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)*.

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.

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

EBERT, D., AND RHEINGANS, P. 2000. Volume Illustration: Non-Photorealistic Rendering of Volume Models. In *Proc. IEEE Visualization*.

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*.

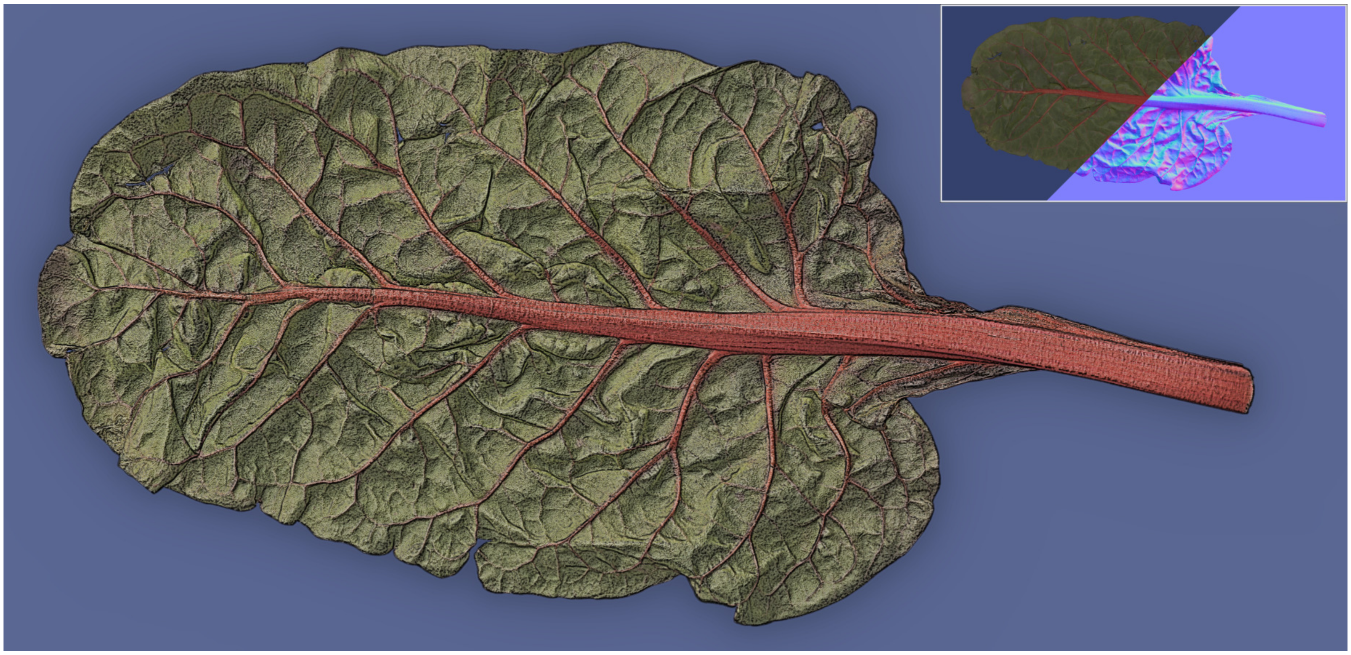


Figure 12: *A leaf of rainbow chard, stylized using a combination of exaggerated shading and suggestive contour lines.*

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

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.

HODGES, E. 2003. *The Guild Handbook of Scientific Illustration*, 2nd ed. Wiley.

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)*.

NEHAB, D., RUSINKIEWICZ, S., DAVIS, J., AND RAMAMOORTHI, R. 2005. Efficiently Combining Positions and Normals for Precise 3D Geometry. *ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 24, No. 3*.

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*.

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

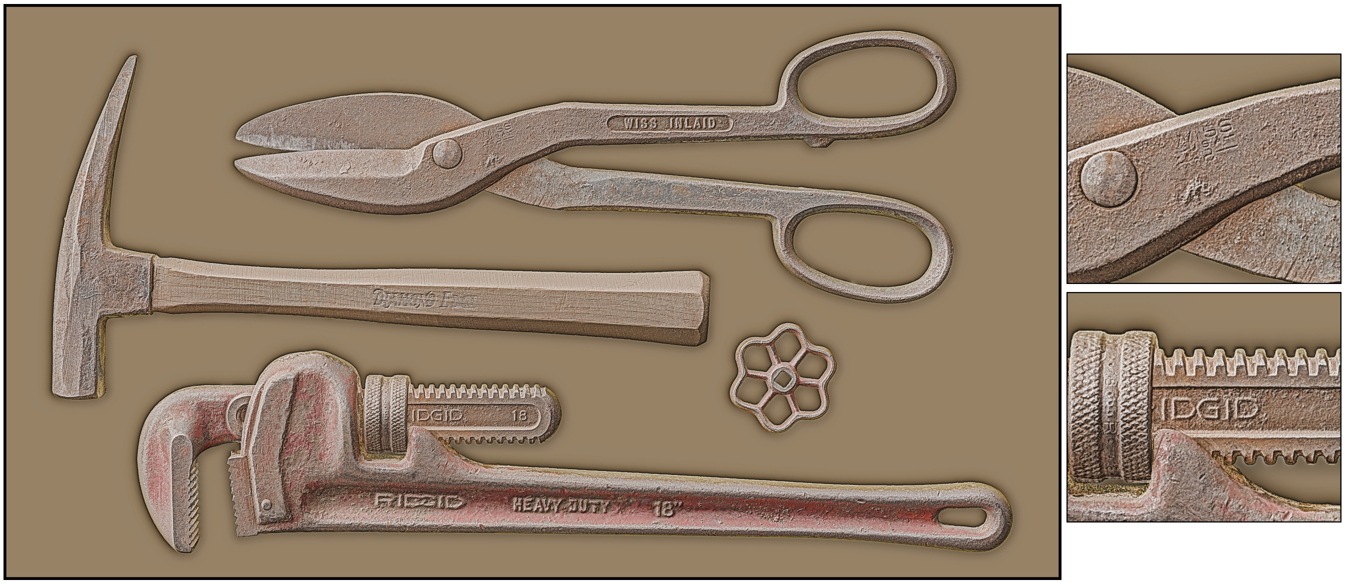


Figure 13: *Illustration of tools reveals fine details, such as the maker's stamp on the shears.*

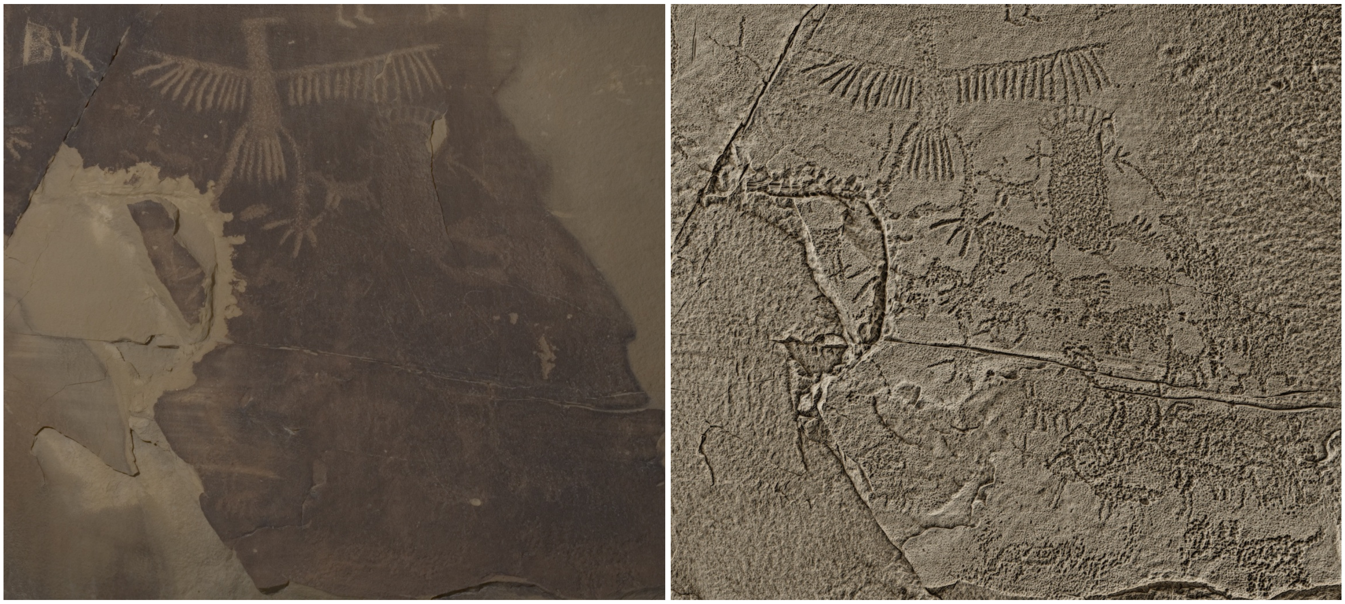


Figure 14: *Color image and nonphotorealistic rendering (with mean curvature shading and exaggerated shading) of the Legend Rock archaeological site.*

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*.

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*.

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*.