Non-photorealistic Camera: Depth Edge Detection and Stylized Rendering using Multi-Flash Imaging

Ramesh Raskar* Kar-Han Tan Rogerio Feris Jingyi Yu† Matthew Turk
Mitsubishi Electric Research Labs (MERL) UC Santa Barbara MIT UC Santa Barbara

Figure 1: *(a) A photo of a car engine (b) Stylized rendering highlighting boundaries between geometric shapes. Notice the four spark plugs and the dip-stick which are now clearly visible (c) Photo of a flower plant (d) Texture de-emphasized rendering.*

Abstract

We present a non-photorealistic rendering approach to capture and convey shape features of real-world scenes. We use a camera with multiple flashes that are strategically positioned to cast shadows along depth discontinuities in the scene. The projective-geometric relationship of the camera-flash setup is then exploited to detect depth discontinuities and distinguish them from intensity edges due to material discontinuities.

We introduce depiction methods that utilize the detected edge features to generate stylized static and animated images. We can highlight the detected features, suppress unnecessary details or combine features from multiple images. The resulting images more clearly convey the 3D structure of the imaged scenes.

We take a very different approach to capturing geometric features of a scene than traditional approaches that require reconstructing a 3D model. This results in a method that is both surprisingly simple and computationally efficient. The entire hardware/software setup can conceivably be packaged into a self-contained device no larger than existing digital cameras.

Keywords: non-photorealistic rendering, image enhancement, depth edges

1 Introduction

Our goal is to create stylized images that facilitate viewer comprehension of the shape contours of the objects depicted. Non-photorealistic rendering (NPR) techniques aim to outline the shapes of objects, highlight the moving parts to illustrate action, and reduce visual clutter such as shadows and texture details [Gooch and Gooch 2001]. The result is useful for imaging low contrast and geometrically complex scenes such as mechanical parts (Figure 1), plants or the internals of a patient (in endoscopy).

When a rich 3D model of the scene is available, rendering subsets of view-dependent contours is a relatively well-understood task in NPR [Saito and Takahashi 1990]. Extending this approach to real scenes by first creating 3D scene models, however, remains difficult. In this paper, we show that it is possible to bypass geometry acquisition, and directly create stylized renderings from images. In the place of expensive, elaborate equipment for geometry acquisition, we propose using a camera with a simple extension: multiple strategically positioned flashes. Instead of having to estimate the full 3D coordinates of points in the scene, and then look for depth discontinuities, our technique reduces the general 3D problem of depth edge recovery to one of intensity step edge detection.

Exploiting the imaging geometry for rendering results in a simple and inexpensive solution for creating stylized images from real scenes. We believe that our camera will be a useful tool for professional artists and photographers, and we expect that it will also enable the average user to easily create stylized imagery.

1.1 Overview

Our approach is based on taking successive photos of a scene, each with a different light source close to and around the camera's center of projection. We use the location of the shadows abutting depth discontinuities as a robust cue to create a depth edge map in both static and dynamic scenes.

Contributions Our main contribution is a set of techniques for detecting and rendering shape contours of scenes with low-contrast or high geometric complexity. Our technical contributions include the following.

- A robust edge classification scheme to distinguish depth edges from texture edges
- A collection of rendering and reconstruction techniques for creating images highlighting shape boundaries from 2D data without creating 3D representations, using qualitative depths
- An image re-synthesis scheme that allows abstraction of textured regions while preserving geometric features
- A technique to detect depth edges in dynamic scenes

*e-mail: [raskar,tan]@merl.com,[rferis,turk]@cs.ucsb.edu
†email: jingyi@graphics.csail.mit.edu

Figure 2: Traditional image enhancement by improving (Left) brightness and (Right) contrast. Low contrast depth edges remain difficult to perceive.

We introduce the concept of a **self-contained stylized imaging device**, a 'non-photorealistic camera', which can directly generate images highlighting contours of geometric shapes in a scene. It contains a traditional camera and embedded flashes, and can be readily and inexpensively built. We attempt to address two important issues in NPR [Gooch and Gooch 2001] [Strothotte and Schlechtweg 2002], detecting shape contours that should be enhanced and identifying features that should be suppressed. We propose a new approach to take image-stylization beyond the processing of a photograph, to actively changing how the photographs are taken.

The output images or videos can be rendered in many ways, e.g., technical illustration, line art or cartoon-like style. We highlight depth discontinuities, suppress material and illumination transitions, and create renderings with large, smoothly colored regions outlined with salient contours [Durand 2002]. We describe several applications: imaging complex mechanical parts, improving images for endoscopes, anatomical drawings and highlighting changes in a scene. Our approach shares the **disadvantages** of NPR: relevant details may be lost as an image is simplified, so tunable abstraction is needed (Section 3.3), and the usefulness of the output is often difficult to quantify.

1.2 Related Work

NPR from images, rather than 3D geometric models has recently received a great deal of attention. The majority of the available techniques for image stylization involve **processing a single image** as the input applying morphological operations, image segmentation, edge detection and color assignment. Some of them aim for stylized depiction [DeCarlo and Santella 2002] [Hertzmann 1998] while others enhance legibility. Interactive techniques for stylized rendering such as rotoscoping have been used as well [Waking Life 2001; Avenue Amy 2002], but we aim to automate tasks where meticulous manual operation was previously required. Our work belongs to an emerging class of techniques to create an enhanced image from multiple images, where the images are captured from the same viewpoint but under different conditions, such as under different illumination, focus or exposure [Cohen et al. 2003; Akers et al. 2003; Raskar et al. 2004].

Aerial imagery techniques find **shadow** evidence by thresholding a single intensity image, assuming flat ground and uniform albedo to infer building heights [Huertas and Nevatia 1988; Irvin and McKeown 1989; Lin and Nevatia 1998]. Some techniques improve shadow capture with novel shadow extraction techniques to compute new shadow mattes [Chuang et al. 2003] or remove them to improve scene segmentation [Toyama et al. 1999]. Some other techniques remove shadows without explicitly detecting them, such as using intrinsic images [Weiss 2001].

Stereo techniques including passive and active illumination are generally designed to compute depth values or surface orientation rather than to detect depth edges. Depth discontinuities present difficulties for traditional stereo: it fails due to *half-occlusions*, i.e., occlusion of scene points in only one of the two views, which confuse the matching process [Geiger et al. 1992]. Few techniques try to model the discontinuities and occlusions directly [Birchfield 1999; Kang et al. 2001; Scharstein and Szeliski 2002]. Active illumination methods, which generally give better results, have been used for depth extraction, shape from shading, shape-time stereo and photometric stereo but are unfortunately unstable around depth discontinuities [Sato et al. 2001]. An interesting technique has been presented to perform logical operations on detected intensity edges, captured under widely varying illumination, to preserve shape boundaries [Shirai and Tsuji 1972] but it is limited to uniform albedo scenes. Using photometric stereo, it is possible to analyze the intensity statistics to detect high curvature regions at **occluding contours** or *folds* [Huggins et al. 2001]. But the techniques assume that the surface is locally smooth which fails for a flat foreground object like a leaf or piece of paper, or view-independent edges such as corner of a cube. They detect regions near occluding contours but not the contours themselves.

Techniques for **shape from shadow** (or darkness) build a continuous representation (*shadowgram*) from a moving light source from which continuous depth estimates are possible [Raviv et al. 1989; Langer et al. 1995; Daum and Dudek 1998]. However, it involves a difficult problem of estimating continuous heights and requires accurate detection of start and end of shadows. Good reviews of shadow-based shape analysis methods are available in [Yang 1996] [Kriegman and Belhumeur 2001] [Savarese et al. 2001].

A common limitation of existing active illuminations methods is that the light sources need to surround the object, in order to create significant shading and shadow variation from (estimated or known 3D) light positions. This necessitates a **fixed lighting rig**, which limits the application of these techniques to industrial settings, and they are impossible to build into a self-contained camera.

We believe our proposed method for extracting depth edges is complementary with many existing methods for computing depth and 3D surface shape, as depth edges often violate smoothness assumptions inherent in many techniques. If the locations of depth discontinuities can be reliably detected and supplied as input, we believe that the performance of many 3D surface reconstruction algorithms can be significantly enhanced.

To find depth edges, we avoid the dependence on solving a correspondence problem or analyzing pixel intensity statistics with moving lights, and we do not attempt to estimate any continuous value. In our search, we have not seen a photometric or other type of stereo method successfully applied to complex scenes where the normals change rapidly– such as a potted plant, or a scene with high depth complexity or low intensity changes, such as a car engine or bone.

1.3 Outline

Our method for creating a stylized image of a static scene consists of the following.

▷ Capture a series of images of the scene under shifted light positions
▷ Process these images to automatically detect depth edges
▷ Identify the subset of intensity edges that are illumination and texture edges
▷ Compute qualitative depth relationships
▷ Enhance or simplify detected features for rendering
▷ Insert processed scene appearance for stylization

We use the term *depth edges* to refer to the C0 discontinuities in a depth map. Depth edges correspond to internal or external occluding contours (or silhouettes) or boundaries of physical objects. The depth edges recovered are *signed*: in the local neighborhood,

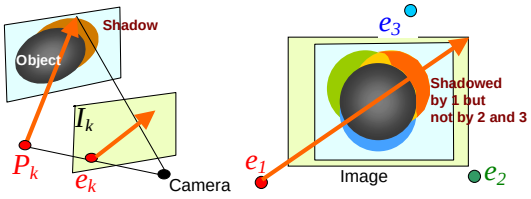

Figure 3: Imaging geometry. Shadows of the gray object are cast along the epipolar ray. We ensure that depth edges of all orientations create shadow in at least one image while the same shadowed points are lit in some other image.

the side with lower depth value, *foreground*, is considered positive while the opposite side is *background* and negative. *Texture edges* are reflectance changes or material discontinuities. Texture edges typically delineate textured regions.

In Section 2, we describe our approach to capturing important features using a multi-flash setup. In Section 3, we discuss methods to use the information to render the images in novel styles. In Section 4, we address the problem of extending the technique to dynamic scenes. We describe our results in Section 5 and conclude with discussion of limitations and future directions.

2 Capturing Edge Features

The image capturing process consists of taking successive pictures of a scene with a point light source *close* to the camera's center of projection (COP). Due to a small *baseline* distance between the camera COP and the light source, a narrow sliver of shadow appears abutting each edge in the image with depth discontinuities; its width depends on the distance from the object edge to the background surface. By combining information about abutting cast shadow from two or more images with distinct light source positions, we can find the depth edges.

2.1 Depth Edges

The method for detecting depth edges is the foundation for our approach. The idea is very simple, in retrospect. It allows us to classify other edges by a process of elimination.

Our method is based on two observations regarding epipolar shadow geometry, as shown in Figure 3. The image of the point light source at P_k is at pixel e_k in the camera image, and is called the *light epipole*. The images of the pencil rays originating at P_k are the *epipolar rays* originating at e_k. (When P_k is behind the camera center, away from the image plane, the epipolar rays wrap around at infinity.) First, note that, a shadow of a depth edge pixel is constrained to lie along the epipolar ray passing through that pixel. Second, the shadow is observed if and only if the background pixel is on the side of the depth edge opposite the epipole *along the epipolar ray*. Hence, in general, if two light epipoles lie on opposite sides of an edge, a cast shadow will be observed at the depth edge in one image but not the other.

We detect shadows in an image by taking a ratio of the image with the maximum composite of all the images. The ratio image accentuates shadows, which abut the depth edges, and de-emphasizes texture edges. During epipolar traversal in the ratio image, the entry point of a shadowed region indicates a depth edge. The basic algorithm is as follows: Given n light sources positioned at $P_1, P_2 \ldots P_n$,

- Capture ambient image I_0
- Capture n pictures I_k^+, $k = 1..n$ with a light source at P_k
- Compute $I_k = I_k^+ - I_0$
- For all pixels x, $I_{max}(x) = \max_k(I_k(x))$, $k = 1..n$
- For each image k,
  - ▷ Create a ratio image, R_k, where $R_k(x) = I_k(x)/I_{max}(x)$
- For each image R_k
  - ▷ Traverse each epipolar ray from epipole e_k
  - ▷ Find pixels y with step edges with negative transition
  - ▷ Mark the pixel y as a depth edge

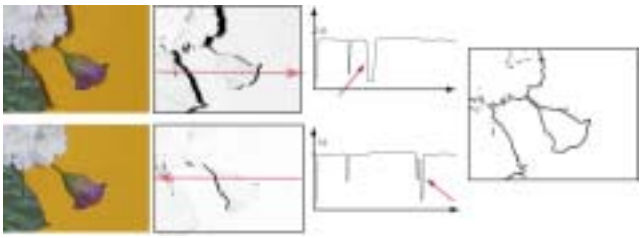

Figure 4: Detecting depth edges. (a) Photo (b) Ratio image (c) Plot along an epipolar ray, the arrows indicate negative transitions (d) Detected edges

With a number of light sources (minimum 2, but typically 4 to 8 are used) placed strategically around the camera, depth edges of all orientation with sufficient depth differences can be detected. In each image, as long as the epipolar ray at a depth edge pixel is not parallel to the image-space orientation of the depth edge, a step edge with negative transition (from lit part to shadowed part) will be detected. If the depth edge is oriented along the epipolar ray, the step edge cannot be detected.

Let us look at the algorithm in detail. Note that, the image I_k has ambient component removed, i.e., $I_k = I_k^+ - I_0$, where I_0 is an image taken with only ambient light and none of the n light sources on. The base image is the maximum composite image, $I_{max.}$, which is an approximation of the image with light source at the camera COP, and in general has no shadows from any of the n light sources. The approximation is close if the n light sources are evenly distributed around the camera COP, have the same magnitude and the baseline is sufficiently smaller than the depth of the scene being imaged.

Consider the image of a 3D point X, given in camera coordinate system, imaged at pixel x. The intensity, $I_k(x)$, if X is lit by the light source at P_k,, under lambertian assumption, is given by

$$I_k(x) = \mu_k \rho(x) \left(\hat{L}_k(x) \cdot N(x)\right)$$

Otherwise, $I_k(x)$ is zero. The scalar μ_k is the magnitude of the light intensity and $\rho(x)$ is the reflectance at X. $\hat{L}_k(x)$ is the normalized light vector $L_k(x) = P_k - X$, and $N(x)$ is the surface normal, all in the camera coordinate system.

Thus, when X is seen by P_k, the ratio is as follows.

$$R_k(x) = \frac{I_k(x)}{I_{\max}(x)} = \frac{\mu_k \left(\hat{L}_x(x) \cdot N(x)\right)}{\max_i \left(\mu_i \left(\hat{L}_i(x) \cdot N(x)\right)\right)}$$

It is clear that, for diffuse objects with nonzero albedo $\rho(x)$, $R_k(x)$ is independent of the albedo $\rho(x)$ and only a function of the local geometry. Further, if the light source-camera baseline$|P_k|$ is small compared to the distance to the point, i.e., $|X| \gg |P_k|$, then this ratio is approximately $\mu_k / \max_i(\mu_i)$, which is a constant for a set of omni-directional light sources in the imaging setup.

The ratio values in $(R_k = I_k/I_{max})$ are close to 1.0 in areas lit by light source k and close to zero in shadowed regions. (In general, the values are not zero due to interreflections). The intensity profile along the epipolar ray in the ratio image shows a sharp negative transition at the depth edge as we traverse from non-shadowed foreground to shadowed background, and a sharp positive transition as we traverse from shadowed to non-shadowed region on the

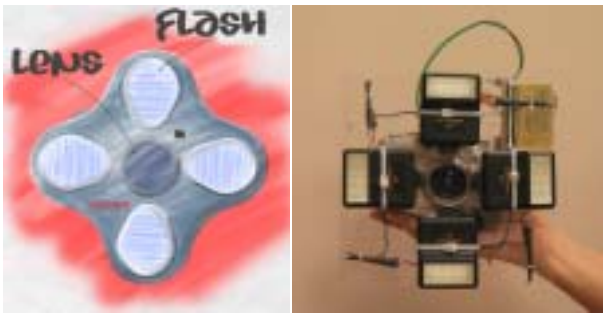

Figure 5: A stylized imaging camera to capture images under four different flash conditions and our prototype.

background (Figure 4). This reduces the depth edge detection problem to an intensity step edge detection problem. A 1D edge detector along the epipolar ray detects both positive and negative transitions, and we mark the negative transitions as depth edges. As mentioned earlier, since we are detecting a transition and not a continuous value, noise and interreflections only affect the accuracy of the position but not the detection of presence of the depth edge.

In summary, there are essentially three steps: (a) create a ratio image where the values in shadowed regions are close to zero; (b) carry out intensity edge detection on each ratio image along epipolar rays marking negative step edges as depth edges (c) combine the edge maps from all n images to obtain the final depth edge map.

Self-contained Prototype An ideal setup should satisfy the constraint that each depth pixel be imaged in both conditions, the negative side of the edge is shadowed at least in one image and not shadowed in at least one other image. We propose using the following configuration of light sources: four flashes at left, right, top and bottom positions (Figure 5).

This setup makes the epipolar ray traversal efficient. If the light source is in the plane parallel to the image plane that contains the center of projection, the light epipole is at infinity and the corresponding epipolar rays are parallel in the image plane. In addition, we place the epipoles such that the epipolar rays are aligned with the camera pixel grid. For the left-right pair, the ray traversal is along horizontal scan lines and for the top-bottom pair, the traversal is along vertical direction.

2.2 Material Edges

In addition to depth edges, we also need to consider illumination and material edges in the image. Illumination edges are boundaries between lit and shadowed regions due to ambient light source(s), rather than the flashes attached to our camera. Since the individual images I_k, are free of ambient illumination, they are free of ambient illumination edges. In general, since material edges are independent of illumination direction, they can be easily classified by a process of elimination. Material edges are intensity edges of I_{max} minus the depth edges.

This edge classification scheme works well and involves a minimal number of parameters for tuning. The only parameters we need are those for intensity edge detection of ratio images and I_{max} image, to detect depth and material edges, respectively.

2.3 Issues

The technique we presented to detect depth edges is surprisingly robust and reliable. We discuss the few conditions in which the basic algorithm fails: a false negative when a negative transition at a

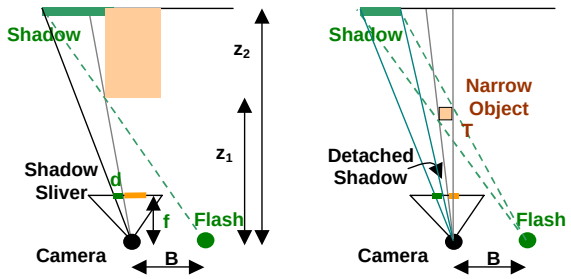

Figure 6: (a) Relationship between baseline and width of shadow (b) Condition where shadow detaches

depth edge cannot be detected in the ratio image R_k or a false positive when other conditions create spurious transitions in R_k. The depth edges can be **missed** due to detached shadows, lack of background, low albedo of background, holes and valleys, or if depth edges lie in shadowed region. The low albedo of background makes it difficult to detect increase in radiance due to a flash, but this problem can be reduced with a higher intensity flash. The problems due to holes/valleys or shadowed depth edges, where the visible background is shadowed for a majority of the flashes, are rare and further reduced when the flash baseline is small. Below, we only discuss the problem due to detached shadows and lack of background. Some pixels may be **mislabeled** as depth edge pixels due to specularities or near silhouettes of curved surfaces. We discuss both these issues. We have studied these problems in detail and the solutions will be provided in a technical report. Here we describe the main ideas.

Curved surfaces The silhouettes on curved surfaces vary smoothly with change in viewpoint and the ratio $R_k(x)$ is very low near depth edges when the 3D contours corresponding to silhouettes with respect to neighboring flash positions are sufficiently different. This is because the dot product$\left(\hat{L}_k(x)\cdot N(x)\right) \approx 0$ and the dot product for light sources on the 'opposite' side will be larger $\left(\hat{L}_i(x)\cdot N(x)\right) > \left(\hat{L}_k(x)\cdot N(x)\right)$. Thus $R_k(x)$ decreases rapidly even though the pixel is not in a shadowed region. However, as seen in examples shown here, this is not a major issue and simply results in a lower slope at the negative transition in R_k. Unlike the problems below, it does not lead to a reversal of intensity gradient along the epipolar ray.

Tradeoff in choosing the baseline A larger baseline distance between the camera and the flash is better to cast a wider detectable shadow in the image, but a smaller baseline is needed to avoid separation of shadow from the associated depth edge.

The width of the abutting shadow in the image is $d = fB(z_2 - z_1)/(z_1 z_2)$, where f is the focal length, B is baseline in

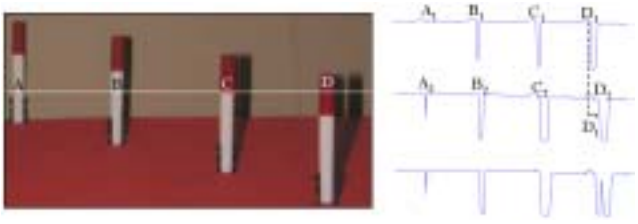

Figure 7: (Left) Minimum composite of image with flash F_S and F_L. (Right) Plot of intensity along a scanline due to F_S, F_L and $\min(I_S, I_L)$.

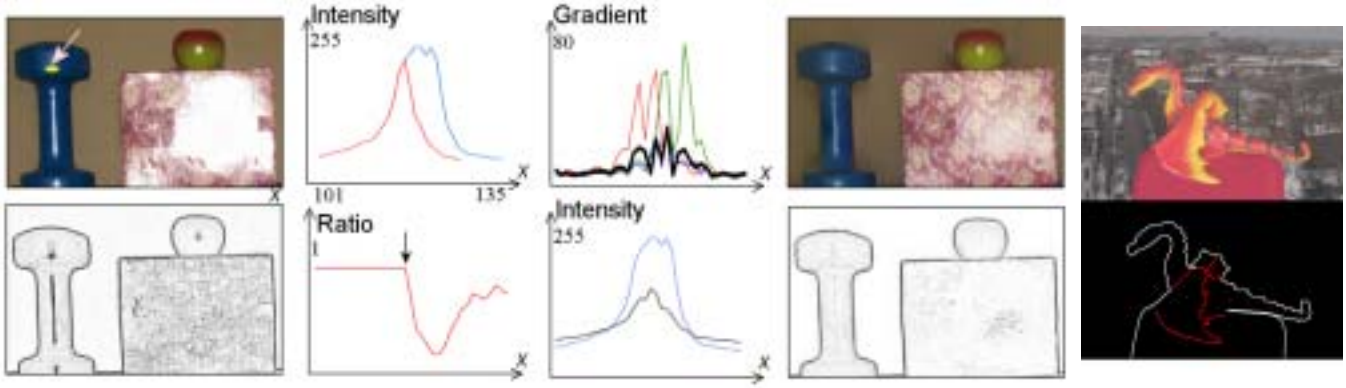

Figure 8: Specularities and lack of background. First column: I_{max} and corresponding result showing artifacts. Second column: For the yellow line marked on dumbbell (x=101:135); Top plot, I_{left} (red) with I_{max} (light blue). Bottom plot, ratio R_{left}. Note the spurious negative transition in R_{left}, at the arrow, which gets falsely identified as a depth edge. Third column: Top plot, gradient of I_{left} (red), I_{right} (green), I_{top} (blue) and Median of these gradients (black). Bottom plot, reconstructed intrinsic image (black) compared with I_{max} (light blue). Fourth column: Top, intrinsic image. Bottom, resulting depth edge map. Fifth column: Top, Scene without a background to cast shadow. Bottom, Edges of I_0/I_{max}, in white plus detected depth edges in red.

mm, and z_1, z_2 are depths, in mm, to the shadowing and shadowed edge. (See Figure 6)

Shadow detachment occurs when the width, T, of the object is smaller than $(z_2 - z_1)B/z_2$. So a smaller baseline, B, will allow narrower objects (smaller T) without shadow separation. Fortunately, with rapid miniaturization and sophistication of digital cameras, we can choose small baseline while increasing the pixel resolution (proportional to f), so that the product fB remains constant, allowing depth detection of narrow objects.

When camera resolutions are limited, we can exploit a **hierarchical baseline** method to overcome this tradeoff. We can detect small depth discontinuities (with larger baselines) without creating shadow separation at narrow objects (using narrow baselines). In practice, we found two different baselines were sufficient. We, however, now have to deal with spurious edges due to shadow separation in the image with larger baseline flash F_L. The image with smaller baseline flash, F_S, may miss small depth discontinuities. How can we combine the information in those two images? There are essentially four cases we need to consider at depth edges (Figure 7) (a) F_S creates a undetectable narrow shadow, F_L creates a detectable shadow (b) F_S creates a detectable small width shadow and F_L creates a larger width shadow. (c) F_S creates detectable shadow but F_L creates a detached shadow that overlaps with F_S shadow and (iv) same as (d) but the shadows of F_S and F_L do not overlap.

Our strategy is based on simply taking the minimum composite of the two images. In the first three cases, this conveniently increases the effective width of the abutting shadow without creating any artifacts, and hence can be treated using the basic algorithm without modifications. For the fourth case, a non-shadow region separates the two shadows in the min composite, so that the shadow in F_L appears spurious.

Our solution is as follows. We compute the depth edges using F_S and F_L (Figure 7). We then traverse the epipolar ray. If the depth edge appears in F_S (at D_1) but not in F_L we traverse the epipolar ray in F_L until the next detected depth edge. If this depth edge in F_L, there is no corresponding depth edge in F_S, we mark this edge as a spurious edge.

The solution using min-composite, however, will fail to detect minute depth discontinuities where even F_L does not create a detectable shadow. It will also fail for very thin objects where even F_S creates a detached shadow.

Specularities Specular highlights that appear at a pixel in one image but not others can create spurious transitions in the ratio images as seen in Figure 8. Although methods exist to detect specularities in a single image [Tan et al. 2003], detecting them reliably in textured regions is difficult.

Our method is based on the observation that specular spots shift according to the shifting of light sources that created them. We need to consider three cases of how specular spots in different light positions appear in each image: (i) shiny spots remain distinct (e.g., on highly specular surface with a medium curvature) (ii) some spots overlap and (iii) spots overlap completely (e.g., on a somewhat specular, fronto-parallel planar surface). Case (iii) does not cause spurious gradients in ratio images.

We note that although specularities overlap in the input images, the boundaries (intensity edges) around specularities in general do not overlap. The main idea is to exploit the gradient variation in the n images at a given pixel (x,y). If (x,y) is in specular region, in cases (i) and (ii), the gradient due to specularity boundary will be high in only one or a minority of the n images under different lighting. The **median of** the n **gradients** at that pixel will remove this outlier(s). Our method is motivated by the intrinsic image approach by [Weiss 2001], where the author removes shadows in outdoor scenes by noting that shadow boundaries are not static. We reconstruct the image by using median of gradients of input images as follows.

- Compute intensity gradient, $G_k(x,y) = \nabla I_k(x,y)$
- Find median of gradients, $G(x,y) = \text{median}_k(G_k(x,y))$
- Reconstruct image I' which minimizes $|\nabla I' - G|$

Image reconstruction from gradients fields, an approximate invertibility problem, is still a very active research area. In R^2, a modified gradient vector field G may not be integrable. We use one of the direct methods recently proposed [Elder 1999] [Fattal et al. 2002]. The least square estimate of the original intensity function, I', so that $G \approx \nabla I'$, can be obtained by solving the Poisson differential equation $\nabla^2 I' = div\ G$, involving a Laplace and a divergence operator. We use the standard full multigrid method [Press et al. 1992] to solve the Laplace equation. We pad the images to square images of size the nearest power of two before applying the integration, and then crop the result image back to the original size [Raskar et al. 2004]. We use a similar gradient domain technique to simplify several rendering tasks as described later.

The resultant intrinsic image intensity, $I'(x,y)$ is used as the denominator for computing the ratio image, instead of the max composite, $I_{max}(x,y)$. In specular regions, the ratio $I_k(x,y)/I'(x,y)$ now is larger than 1.0. This is clamped to 1.0 so that the negative transitions in the ratio image do not lie in specular parts.

Figure 9: (a) A edge rendering with over-under style. (b) Rendering edges with width influenced by orientation. (c) and (d) Normal Interpolation for toon rendering exploiting over-under mattes.

Lack of Background Thus far we assumed that depth edges casting shadows on a background are within a finite distance. What if the background is significantly far away or not present? This turns out to be a simple situation to solve because in these cases only the outermost depth edge, the edge shared by foreground and distant background, is missed in our method. This can be easily detected with a foreground-background estimation technique. In I_{max} image the foreground pixels are lit by at least one of the flashes but in the ambient image, I_0, neither the foreground nor the background is lit by any flash. Hence, the ratio of I_0/I_{max}, is near 1 in background and close to zero in interior of the foreground. Figure 8 shows intensity edges of this ratio image combined with internal depth edges.

3 Image Synthesis

Contour-based comprehensible depiction is well explored for 3D input models [DeCarlo et al. 2003] but not for photographs. In the absence of a full 3D representation of the scene, we exploit the following 2D cues to develop novel rendering algorithms.

(a) The sign of the depth edge,
(b) Relative depth difference based on shadow width,
(c) Color near the signed edges, and
(d) Normal of a smooth surface at the occluding contour

We aim to automate tasks for stylized rendering where meticulous manual operation was originally required, such as image editing or rotoscoping [Waking Life 2001] .

3.1 Rendering Edges

We create a vectorized polyline representation of the depth edges by linking the depth edge pixels into a contour. The polyline is smoothed and allows us to stylize the width and color of the contour maintaining spatial coherency. While traversing the marked depth edge pixels to create a contour, at T-junctions, unlike traditional methods that choose the next edge pixel based on orientation similarity, we use the information from the shadows to resolve the connected component. Two edge pixel are connected only if they are connected in the intensity edges of all the n ratio images.

Signed edges At the negative transition along the epipolar ray in the ratio image, R_k, the side of edge with higher intensity is the foreground and lower intensity (corresponding to shadowed region) is background. This qualitative depth relationship can be used to clearly indicate foreground-background separation at each edge. We emulate the over-under style used by artists in mattes. The foreground side is white while the background side is black. Both are rendered by displacing depth contour along the normal (Figure 9(a)).

Light direction We use a commonly known method to convey light direction by modifying the width of edges depending on the edge orientation. Since the edge orientation in 3D is approximately the same as the orientation of its projection in image plane, the thickness is simply proportional to the dot product of the image space normal with a desired light direction (Figure 9(b)).

Color variation We can indicate color of original object by rendering the edges in color. From signed edges, we pick up a foreground color along the normal at a fixed pixel distance, without crossing another depth or intensity edge. The foreground colored edges can also be superimposed onto a segmented source image as seen in Figure 10(c).

3.2 Color Assignment

Since there is no 3D model of the scene, rendering non-edge pixels requires different ways of processing captured 2D images.

Normal interpolation For smooth objects, the depth edge corresponds to the occluding contour where the surface normal is perpendicular to the viewing direction. Hence the normals at depth edges lie in the plane of the image and we can predict normals at other pixels. We solve this sparse interpolation problem by solving a 2D Poisson differential equation. Our method is inspired by the Lumo [Johnston 2002] where the over-under mattes are manually created. In our case, signed depth edges allow normal interpolation while maintaining normal discontinuity at depth edges.

Image attenuation We accentuate the contrast at shape boundaries using an image attenuation maps (Figure 10(a)) as follows. Depth edges are in white on a black background. We convolve with a filter that is the gradient of an edge enhancement filter. Our filter is a Guassian minus an impulse function. When we perform a 2D integration on the convolved image, we get a sharp transition at the depth edge.

Depicting Change Some static illustrations demonstrate action e.g., changing oil in a car, by making moving parts in the foreground brighter. Foreground detection via intensity-based schemes, however, is difficult when the colors are similar and texture is lacking, e.g., detecting hand gesture in front of other skin colored parts (Figure 11). We take two separate sets of multi-flash shots, without and with the hand in front of the face to capture the reference and changed scene. We note that any change in a scene is bounded by new depth edges introduced. Without explicitly detecting foreground, we highlight interiors of regions that contribute to new depth edges.

We create a gradient field where pixels marked as depth edges in changed scene but not in reference, are assigned a unit magnitude gradient. The orientation matches the image space normal to the depth edge. The gradient at other pixels is zero. The reconstructed image from 2D integration is a pseudo-depth map – least squared error solution via solving Poisson equation. We threshold this map at 1.0 to get the foreground mask which is brightened. Note, the shadow width along the epipolar ray is proportional to the ratio of depth values on two sides of the edge. Hence instead of a unit magnitude gradient, we could assign a value proportional

Figure 10: Color assignment. (a) Attenuation Map (b) Attenuated Image (c) Colored edges on de-emphasized texture

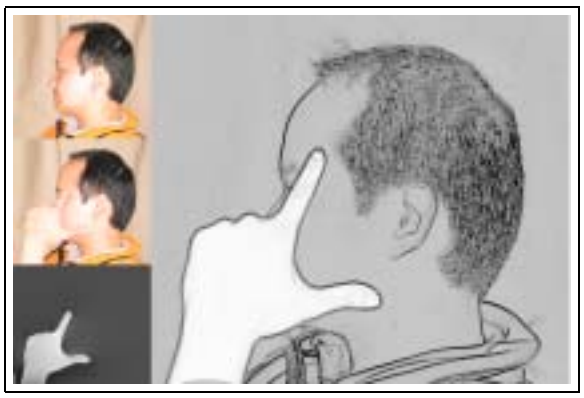

Figure 11: Change Detection. (Left column) Reference image, changed image, and pseudo depth map of new depth edges (Right) Modified depth edge confidence map.

to the logarithm of the shadow width along the epipolar ray to get a higher quality pseudo-depth map. Unfortunately, we found that the positive transition along the ray is not strong due to the use of a non-point light source and interreflections. In principle, estimated shadow widths could be used for say, tunable abstraction to eliminate edges with small depth difference.

3.3 Abstraction

One way to reduce visual clutter in an image and emphasize object shape is to simplify details not associated with the shape boundaries (depth edges) of the scene, such as textures and illumination variations [Gooch and Gooch 2001]. Our goal is to create large flat colored regions separated by strokes denoting important shape boundaries. Traditional NPR approaches based on image segmentation achieve this by assigning a fixed color to each segment [DeCarlo and Santella 2002]. However, image segmentation may miss a depth edge leading to merger of foreground and background near this edge into a single colored object. Although image segmentation can be guided by the computed depth edges, the segmentation scheme places hard constraint on closed contours and does not support smalls gaps in contours. We propose a method that is conceptually simple and easy to implement.

Our method reconstructs image from gradients without those at texture pixels. No decision need to be made about what intensity values to use to fill in holes, and no feathering and blurring need be done, as is required with conventional pixel-based systems. We use a mask image, γ, to attenuate the gradients away from depth edges. The mask image is computed as follows.

$\gamma(x,y) = a$ if (x,y) is a texture edge pixel
$= a \cdot d(x,y)$ if (x,y) is a featureless pixel
$= 1.0$ if (x,y) is a depth edge pixel

The factor $d(x,y)$ is the ratio of the distance field of texture pixels by the distance field of depth edge pixels. The distance field value at a pixel is the Euclidean distance to the nearest (texture or depth) edge pixel. As shown in Figure 12, the parameter a controls the degree of abstraction, and textures are suppressed for $a = 0$. The procedure is as follows.

- Create a mask image $\gamma(x,y)$
- Compute intensity gradient $\nabla I(x,y)$
- Modify masked gradients $G(x,y) = \nabla I(x,y)^{\gamma(x,y)}$
- Reconstruct image I' to minimize $|\nabla I' - G|$
- Normalize $I'(x,y)$ colors to closely match $I(x,y)$

The image reconstruction follows the solution of a Poisson equation via a multi-grid approach as in the specularity attenuation technique in Section 2.

Figure 12: Tunable abstraction for texture de-emphasis. Depth edge followed by abstraction with $a = 1$, $a = 0.5$ and $a = 0$.

4 Dynamic Scenes

Our method for capturing geometric features thus far requires taking multiple pictures of the same static scene. We examine the **lack of simultaneity** of capture for scenes with moving objects or a moving camera. Again, a large body of work exists for estimating motion in image sequences, and a sensible approach is to use the results from the static algorithm and apply motion compensation techniques to correct the artifacts introduced. Finding optical flow and motion boundaries, however, is a challenging problem especially in textureless regions [Papademetris and Belhumeur 1996; Birchfield 1999]. Fortunately, by exploiting properties of our unique imaging setup, in most cases, movement of depth edges in dynamic scenes can still be detected by observing the corresponding movement in shadowed regions. As in the static case, we bypass

the hard problem of finding the rich per-pixel motion representation and focus directly on finding the discontinuities i.e., depth edges in motion. The setup is similar to the static case with n flashes around the camera, but triggered in a rapid cyclic sequence, one flash per frame. We find depth edges in a given frame and connect edges found in adjacent frames into a complete depth edge map.

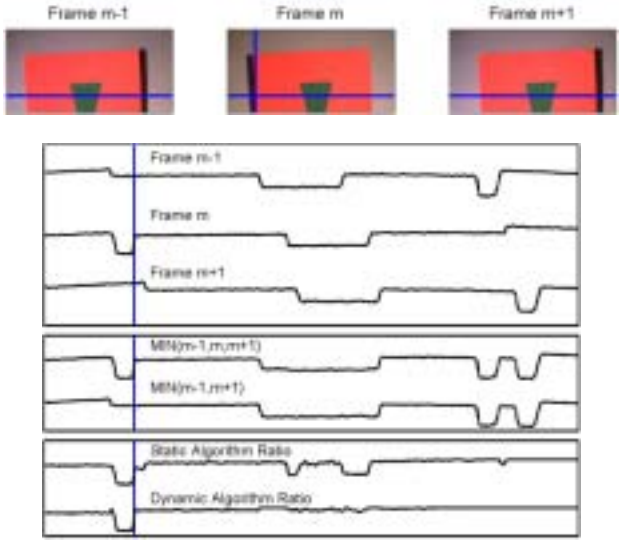

Figure 13: Depth edge detection for dynamic scenes. (Top) Three frames from multi-flash sequence of a toy example showing a red square with a green triangle texture moving from left to right. We are interested in detecting the depth edge in frame m. A single scan line shown in blue is used for the plots. (Middle) The three scan lines plots. The position of the correct depth edge position is indicated with a vertical blue line. (Bottom) Plot of minimum composite and ratio images computed using the static and dynamic algorithms. The motion induced unwanted edges in the static ratio image but not in the dynamic ratio image. The correct depth edge can then be detected from the ratio image using the same traversal procedure as before.

4.1 Depth Edges in Motion

To simplify the discussion, consider using just the left and right flashes to find vertical depth edges. Images from three frames, I_{m-1}, I_m and I_{m+1}, from a toy example are shown in Figure 13. In the sequence, a red square with a green triangle texture is shown moving from left to right, and the three frames are captured under left, right, and left flashes, as can be easily inferred from the cast shadows.

In presence of scene motion, it is difficult to reliably find shadow regions since the base image to compare with, e.g., the max composite, I_{max}, exhibits misaligned features. A high speed camera can reduce the amount of motion between frames but the lack of simultaneity cannot be assumed.

We make two simplifying assumptions (a) motion in image space is monotonic during the image capture from the start of frame m-1 to the end of frame m+1 and (b) the motion is also small enough that the depth and texture edges in the frames do not cross, i.e., the motion is restricted to the spacing between adjacent edges on the scan line.

Due to the left-right switch in illumination, a shadow near a depth edge disappears in alternate frame images, I_{m-1} and I_{m+1}, while a moving texture edge appears in all three frames. Monotonicity of motion without crossing over edges means $\min(I_{m-1}, I_{m+1})$ or $\max(I_{m-1}, I_{m+1})$ will both have a flat region around the depth edge in frame m. Similarly, images $\min(I_{m-1}, I_m, I_{m+1})$ and $\max(I_{m-1}, I_m, I_{m+1})$ both are bound to have a flat region around texture edge in frame m. Since the cast shadow region at the depth edge in frame m is darker than the foreground and background objects in the scene, the shadow is preserved in $\min(I_{m-1}, I_m, I_{m+1})$ but not in $\max(I_{m-1}, I_m, I_{m+1})$. This leads to the following algorithm:

- Compute shadow preserving $I_t = \min(I_{m-1}, I_m, I_{m+1})$
- Compute shadow free $I_d = \min(I_{m-1}, I_{m+1})$
- Compute ratio image, R_m, where $R_m = I_t / I_d$
- Traverse along epipolar ray from e_m and mark negative transition

This ratio image is free of unwanted transitions and the same epipolar ray traversal method can be applied to localize the depth edges.

Figure 13 shows the algorithm in action. We tested the algorithm with synthetic sequences to investigate the set of conditions under which the algorithm is able to correctly localize the depth edges and also experimented with this algorithm in real dynamic scenes. An example frame from a dynamic sequence is shown in Figure 14. A full stylized example with human subjects can be seen in the accompanying video. While we are very encouraged by the simplicity of the algorithm as well as the results we were able to achieve with it, the simplifying assumptions made about the monotonicity and magnitude of motion are still fairly restrictive. For thin objects or objects with high frequency texture, large motions between successive frames creates spurious edges. We plan to continue our investigation in this area and designing algorithms that require fewer assumptions and work under a wider range of conditions.

Figure 14: (Left) A frame from a video sequence, shadows due to left flash. (Right) Detected depth edges merged from neighboring frames.

4.2 Edges and Colors

The depth edges in a given frame, m, are incomplete since they span only limited orientations. In a dynamic scene a union of depth edges from all n successive frames may not line up creating discontinuous contours. We match signed depth edges corresponding to the same flash i.e., m and $m+n$ and interpolate the displacement for intermediate frames. To assign colors, we take the maximum of three successive frames. Our video results can also be considered as tools for digital artists who traditionally use rotoscoping for finding shape boundaries in each frame.

5 Implementation

Our basic prototype makes use of a 4 MegaPixel Canon Powershot G3 digital camera. The dynamic response in the images is linearized. The four booster (slaved Quantarray MS-1) 4ms duration flashes are triggered by optically coupled LEDs turned on sequentially by a PIC microcontroller, which in turn is interrupted by the

hot-shoe of the camera. Our video camera is a PointGrey DragonFly camera at 1024x768 pixel resolution, 15 fps which drives the attached 5W LumiLeds LED flashes in sequence. We used a *Lumina Wolf* endoscope with 480x480 resolution camera.

It takes 2 seconds to capture each image. Our basic algorithm to detect depth edges executes in 5 seconds in C++ on a Pentium4 3GHz PC. The rendering step for 2D Poisson takes about 3 minutes.

6 Results

We show a variety of examples of real scenes, from millimeter scale objects to room sized environments.

Figure 15: Room sized scene: Right flash image and depth edge map.

Objects and room sized scenes We examine imaging a mechanical (car engine, Figure 1(b)), organic (plant, Figure 1(d)) and anatomical (bone, Figure 9) object. For organic objects, such as flower plant, the geometric shape is complex with specular highlights, probably challenging for many shape-from-x algorithms. Note the individual stems and leafs that are clear in the new synthesis. The white **bone** with complex geometry, is enhanced with different shape contour styles. In all these scenes, intensity edge detection and color segmentation produce poor results because the objects are almost uniformly colored. The method can be easily used with room-sized scenes (Figure 15).

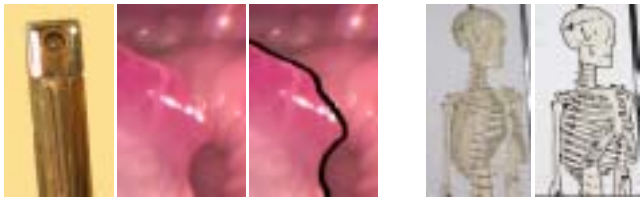

Figure 16: (Left) Enhanced endoscope, with only left lights turned on; input image and depth edge superimposed image. (Right) Skeleton and depth edge superimposed image.

Milli-scale Scene Medical visualization can also benefit from multi-flash imaging. We manipulated the two light sources available near the tip of an endoscopic camera. The baseline is 1mm for 5mm wide endoscope (Figure 16.left). From our discussions with medical doctors and researchers who with such images, extension to video appears to be a promising aid in examination [Tan et al. 2004]. A similar technique can also be used in boroscopes that are used to check for gaps and cracks inside inaccessible mechanical parts - engines or pipes.

Comparison with other strategies We compared our edge rendering technique for comprehension with intensity edge detection using Canny operator, and segmentation. We also compared with active illumination stereo 3D scanning methods, using a state of the art 3Q scanner. Edges captured via **intensity edge detection** are sometimes superimposed on scenes to improve comprehension. While this works in high contrast imagery, sharp changes in image values do not necessarily imply object boundaries, and vice versa [Forsyth and Ponce 2002]. The Canny edge detection or segmentation based NPR approaches unfortunately also fail in low-contrast areas e.g., in the plant, bone or engine (Figure 17.left) example. The 3D scanner output is extremely high quality in the interior of objects as well as near the depth edges. But due to partial occlusions, the depth edges are noisy (Figure 17).

Figure 17: (Left) Intensity edge detection (Canny) for engine of Figure 1(a). (Right Top) Depth map from 3Q scanner, notice the jagged depth edges on the neck. (Right Bottom) Depth edge confidence map using our technique.

7 Discussion

Feature capture For comprehensible imagery, **other shape cues** such as high curvature regions (ridges, valleys and creases) and self-shadowing boundaries from external point light sources are also useful, and are not captured in our system. Our method is highly dependent on being able to detect the scene radiance contributed by the flash, so bright outdoors or distant scenes are a problem. Given the dependence on shadows of opaque objects, our method cannot handle transparent, translucent, luminous, and mirror like objects.

Many **hardware improvements** are possible. Note that the depth edge extraction scheme could be used for spectrums other than visible light that create 'shadows', e.g., in infrared, sonar, X-rays and radars imaging. Specifically, we envision the video-rate camera to be fitted with infrared light sources invisible to humans so the resulting flashes are not distracting. In fact, one can use a frequency division multiplexing scheme to create a **single shot** multi-flash photography. The flashes simultaneously emit four different colors (wavelength) and the Bayer mosaic like pattern of filters on the camera imager decodes the four separate wavelengths.

Applications of depth edges Detecting depth discontinuity is fundamental to image understanding and can be used in many applications [Birchfield 1999]. Although current methods rely primarily on outermost silhouettes of objects, we believe a complete depth edge map can benefit problems in visual hull, segmentation, layer resolving and aspect graphs. Aerial imaging techniques [Lin and Nevatia 1998] can improve building detection by looking at *multiple* time-lapsed images of cast shadows from known sun directions before and after local noon. In addition, effects such as depth of field effect during post-processing, synthetic aperture using camera array and screen matting for virtual sets (with arbitrary background) require high quality *signed* depth edges.

Edge-based or area-based stereo correspondence can be improved by matching signed depth edges, constraining dynamic pro-

gramming to segments within depth edges and modifying correlation filters to deal with partial occlusions [Scharstein and Szeliski 2002]. Edge classification can provide confidence map to assist color and texture segmentation in low-contrast images. Shape contours can also improve object or gesture recognition [Feris et al. 2004].

8 Conclusion

We have presented a simple yet effective method to convey shape boundaries by rendering new images and videos of real world scenes. We exploit the epipolar relationship between light sources and cast shadows to extract geometric features from multiple images of a scene. By making use of image space discontinuity rather than relying on 3D scene reconstruction, our method can robustly capture the underlying primitives for rendering in different styles.

We have presented basic prototypes, related feature capturing and rendering algorithms, and demonstrated applications in technical illustration and video processing. Finally, since a depth edge is such a basic primitive, we have suggested ways in which this information can be used in applications beyond NPR.

Minor modification to camera hardware enables this method to be implemented in a self-contained device no larger than existing digital cameras. We have proposed one possible approach to leveraging the increasing sophistication of digital cameras to easily produce useful and interesting stylized images.

Acknowledgements We thank the anonymous reviewers for useful comments and guidance. We thank Adrian Ilie, Hongcheng Wang, Rebecca Xiong, Paul Beardsley, Darren Leigh, Paul Dietz, Bill Yerazunis and Joe Marks for stimulating discussions, James Kobler (MEEI), Takashi Kan and Keiichi Shiotani for providing motivating applications, Narendra Ahuja and Beckman Institute Computer Vision and Robotics Lab for suggestions and support, and many members of MERL for help in reviewing the paper.

References

AKERS, D., LOSASSO, F., KLINGNER, J., AGRAWALA, M., RICK, J., AND HANRAHAN, P. 2003. Conveying Shape and Features with Image-Based Relighting. In *IEEE Visualization*.

AVENUE AMY, 2002. Curious Pictures.

BIRCHFIELD, S. 1999. *Depth and Motion Discontinuities*. PhD thesis, Stanford University.

CHUANG, Y.-Y., GOLDMAN, D. B., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2003. Shadow matting and compositing. *ACM Trans. Graph. 22*, 3, 494–500.

COHEN, M. F., COLBURN, A., AND DRUCKER, S. 2003. Image stacks. Tech. Rep. MSR-TR-2003-40, Microsoft Research.

DAUM, M., AND DUDEK, G. 1998. On 3-D Surface Reconstruction using Shape from Shadows. In *CVPR*, 461–468.

DECARLO, D., AND SANTELLA, A. 2002. Stylization and Abstraction of Photographs. In *Proc. Siggraph 02, ACM Press*.

DECARLO, D., FINKELSTEIN, A., RUSINKIEWICZ, S., AND SANTELLA, A. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph. 22*, 3, 848–855.

DURAND, F. 2002. An Invitation to Discuss Computer Depiction. In *Proceedings of NPAR 2002*.

ELDER, J. 1999. Are Edges Incomplete? . *International Journal of Computer Vision 34*, 2/3, 97–122.

FATTAL, R., LISCHINSKI, D., AND WERMAN, M. 2002. Gradient Domain High Dynamic Range Compression. In *Proceedings of SIGGRAPH 2002*, ACM SIGGRAPH, 249–256.

FERIS, R., TURK, M., RASKAR, R., TAN, K., AND OHASHI, G. 2004. Exploiting Depth Discontinuities for Vision-based Fingerspelling Recognition. In *IEEE Workshop on Real-time Vision for Human-Computer Interaction (in conjunction with CVPR'04)*.

FORSYTH, AND PONCE. 2002. *Computer Vision, A Modern Approach*.

GEIGER, D., LADENDORF, B., AND YUILLE, A. L. 1992. Occlusions and Binocular Stereo. In *European Conference on Computer Vision*, 425–433.

GOOCH, B., AND GOOCH, A. 2001. *Non-Photorealistic Rendering*. A K Peters, Ltd., Natick.

HERTZMANN, A. 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In *ACM SIGGRAPH*, 453–460.

HUERTAS, A., AND NEVATIA, R. 1988. Detecting buildings in aerial images. *Computer Vision, Graphics and Image Processing 41*, 2, 131–152.

HUGGINS, P., CHEN, H., BELHUMEUR, P., AND ZUCKER, S. 2001. Finding Folds: On the Appearance and Identification of Occlusion . In *IEEE CVPR*, vol. 2, 718–725.

IRVIN, R., AND MCKEOWN, D. 1989. Methods for exploiting the relationship between buildings and their shadows in aerial imagery. *IEEE Transactions on Systems, Man and Cybernetics 19*, 6, 1564–1575.

JOHNSTON, S. F. 2002. Lumo: Illumination for cel animation. In *Proceedings of NPAR*, ACM Press, 45–52.

KANG, S. B., SZELISKI, R., AND CHAI, J. 2001. Handling occlusions in dense multi-view stereo. In *IEEE CVPR*, vol. 1, 102–110.

KRIEGMAN, D., AND BELHUMEUR, P. 2001. What Shadows Reveal About Object Structure . *Journal of the Optical Society of America*, 1804–1813.

LANGER, M., DUDEK, G., AND ZUCKER, S. 1995. Space Occupancy using Multiple Shadow Images. *International Conference on Intelligent Robots and Systems*, 390–396.

LIN, C., AND NEVATIA, R. 1998. Building detection and description from a single intensity image. *Computer Vision and Image Understanding: CVIU 72*, 2, 101–121.

PAPADEMETRIS, X., AND BELHUMEUR, P. N. 1996. Estimation of motion boundary location and optical flow using dynamic programming. In *Proc. Int. Conf. on Image Processing*.

PRESS, W. H., TEUKOLSKY, S., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C: The Art of Scientific Computing* . Pearson Education.

RASKAR, R., ILIE, A., AND YU, J. 2004. Image Fusion for Context Enhancement and Video Surrealism. In *Proceedings of NPAR*.

RAVIV, D., PAO, Y., AND LOPARO, K. A. 1989. Reconstruction of Three-dimensional Surfaces from Two-dimensional Binary Images. In *IEEE Transactions on Robotics and Automation*, vol. 5(5), 701–710.

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

SATO, I., SATO, Y., AND IKEUCHI, K. 2001. Stability issues in recovering illumination distribution from brightness in shadows. *IEEE Conf. on CVPR*, 400–407.

SAVARESE, S., RUSHMEIER, H., BERNARDINI, F., AND PERONA, P. 2001. Shadow Carving. In *ICCV*.

SCHARSTEIN, D., AND SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *International Journal of Computer Vision*, vol. 47(1), 7–42.

SHIRAI, Y., AND TSUJI, S. 1972. Extraction of the Line Drawing of 3-Dimensional Objects by Sequential Illumination from Several Directions. *Pattern Recognition 4*, 4, 345–351.

STROTHOTTE, T., AND SCHLECHTWEG, S. 2002. *NonPhotorealistic Computer Graphics: Modeling, Rendering and Animation*. Morgan Kaufmann, San Francisco.

TAN, P., LIN, S., QUAN, L., AND SHUM, H.-Y. 2003. Highlight Removal by Illumination-Constrained Inpainting . In *Ninth IEEE International Conference on Computer Vision*.

TAN, K., KOBLER, J., DIETZ, P., FERIS, R., AND RASKAR, R. 2004. Shape-Enhanced Surgical Visualizations and Medical Illustrations with Multi-Flash Imaging. In *MERL TR/38*.

TOYAMA, K., KRUMM, J., BRUMITT, B., AND MEYERS, B. 1999. Wallflower: Principles and Practice of Background Maintenance. In *ICCV*, 255–261.

WAKING LIFE, 2001. Waking Life, the movie.

WEISS, Y. 2001. Deriving intrinsic images from image sequences. In *Proceedings of ICCV*, vol. 2, 68–75.

YANG, D. K.-M. 1996. *Shape from Darkness Under Error*. PhD thesis, Columbia University.