Dealing with Multi-Scale Depth Changes and Motion in Depth Edge Detection

Rogerio Feris Matthew Turk
University of California, Santa Barbara

Ramesh Raskar
Mitsubishi Electric Research Labs

Abstract

Sharp discontinuities in depth, or depth edges, are very important low-level features for scene understanding. Recently, we have proposed a solution to the depth edge detection problem using a simple modification of the capture setup: a multi-flash camera with flashes appropriately positioned to cast shadows along depth discontinuities in the scene. In this paper, we show that by varying illumination parameters, such as the number, spatial position, and wavelength of light sources, we are able to handle fundamental problems in depth edge detection, including multi-scale depth changes and motion. The robustness of our methods is demonstrated through our experimental results in complex scenes.

1. Introduction

A great deal of computer vision tasks rely on intensity edges as low-level features. Edge-based methods, however, are limited to reveal scene structure: many sharp intensity transitions are produced by texture or illumination variations that are not aggregated along geometric curves. In addition, important gray level discontinuities along occlusion boundaries might have low contrast or appear blurred due to the imaging process. As a result, intensity edge maps may include undesirable edges (due to albedo changes, specularities or shadows), or miss edges along important shape boundaries (see Figure 8b).

Ideally, we should describe discontinuities in the physical surfaces rather than edges in the image intensities. The latter are somewhat arbitrary and do not always correspond to physical properties of surfaces; the former are well-defined and real [7].

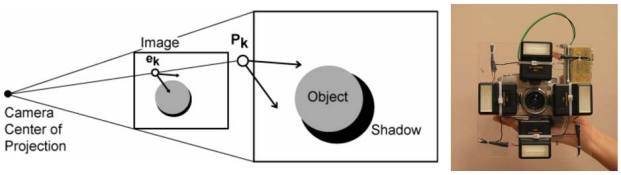
Our work addresses the problem of detection of discontinuities in depth, also known as depth edges or occluding contours. Depth edges are directly related to the 3D scene geometry and provide extremely important low-level features for image understanding, since they tend to outline the boundaries of objects in the scene. In fact, they comprise one of the four components in the well-known 2 1/2-D sketch of Marr's computational vision model [10].

Reliable detection of depth edges clearly facilitates segmentation, establishes depth-order relations, and provides valuable features for visual recognition [5], tracking, and 3D reconstruction [3]. It can also be used for camera control, and non-photorealistic rendering [12].

Most previous approaches proposed for detection of depth discontinuities treat them as an annoyance, rather than as a positive source of information [2]. The reason is that the majority of 3D reconstruction methods produce inaccurate results near depth discontinuities, due to occlusions and the violation of smoothness constraints. Recently, steady progress has been made in discontinuity preserving stereo matching [1], mainly with global optimization algorithms based on belief propagation or graph cuts (see [16] for a comparison). However, these methods fail to capture depth edges associated with sufficiently small changes in depth. Moreover, obtaining clean, non-jagged contours along shape boundaries is still a challenging problem even for methods that rely on more expensive hardware [8].

Instead of having to estimate the full 3D coordinates of points in the scene, and then look for depth discontinuities, our technique bypasses geometry acquisition, and directly detects depth edges. Recently, we have proposed a method that relies on a simple modification of the capture setup: a multi-flash camera is used with flashes appropriately positioned to cast shadows along depth discontinuities in the scene. We have demonstrated applications in non-photorealistic rendering [12], fingerspelling recognition [5], specular reflection reduction [4], medical imaging [15], and stereo vision [3]. The main contributions of this paper are listed below:

- We exploit multi-flash imaging in a more general way, showing that by varying illumination parameters, such as the number, spatial position, and wavelength of light sources, we are able to better approach the problem of depth edge detection. In this sense, our previous work (described in Section 2) is a particular instance of the framework presented in this paper.
- Within this multi-flash imaging framework, we provide a detailed analysis of the problem of multi-scale depth changes (Section 3), which was briefly introduced in our previous work [12].
- A novel algorithm for detecting depth edges in dynamic scenes is then proposed, relying on the wavelength variation of the light sources. (Section 4).



(a) (b)

Figure 1: *(a) Imaging Geometry. (b) Our implementation setup with four flashes.*

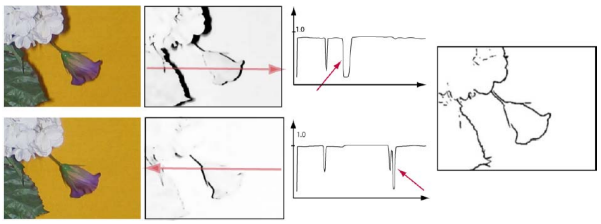


Figure 2: *From left to right: photo, ratio image, plot along an epipolar ray (the arrow indicates negative transitions) and detected edges.*

2. Depth Edges with Multi-Flash

Our method to capture shape features [12] is motivated by the observation that when a flash (*close* to the camera) illuminates a scene during image capture, thin slivers of cast shadow are created at depth discontinuities. Thus, if we can shoot a sequence of images in which different light sources illuminate the subject from various positions, we can use the shadows in each image to assemble a depth edge map using the shadow images.

In order to capture the intuitive notion of how the position of the cast shadows is dependent on the relative position of the camera and light sources, we examine the imaging geometry, illustrated in Figure 1a. Adopting a pinhole camera model, the projection of the point light source at P_k is at pixel e_k on the imaging sensor. We call this *image* of the light source the *light epipole*. The images of (the infinite set of) light rays originating at P_k are in turn called the *epipolar rays*, originating at e_k.

There are two simple observations that can be made about cast shadows: (1) a shadow of a depth edge pixel is constrained to lie along the epipolar ray passing through that pixel and (2) when a shadow is induced at a depth discontinuity, the shadow and the light epipole will be at opposite sides of the depth edge.

These two observations suggest that if we can detect shadow regions in an image, then depth edges can be localized by traversing the epipolar rays starting at the light epipole and identifying the points in the image where the shadows are first encountered.

Our approach for reliably detecting shadows in the images is to position lights so that every point in the scene that is shadowed in some image is also captured without being shadowed in at least one other image. This can be achieved by placing lights appropriately so that for every light, there is another on the opposite side of the camera to ensure that all depth edges are illuminated from two sides.

To detect shadows in each image, we first compute a *shadow-free image*, which can be approximated with the maximum composite image, which is an image assembled by choosing at each pixel the maximum intensity value among the image set. The shadow-free image is then compared with the individual shadowed images. In particular, for each shadowed image, we compute the *ratio image* by performing a pixel-wise division of the intensity of the shadowed image by the intensity of the maximum image. The ratio image is close to 1 at pixels that are not shadowed, and close to 0 at pixels that are shadowed. This serves to accentuate the shadows and remove intensity transitions due to surface material changes.

The final step of our algorithm is to traverse the ratio images along the correspondent epipolar rays and mark negative transitions as depth edges. In our implementation, we used four flashes at left, right, top and bottom positions (see Figure 1b). This setup makes the epipolar ray traversal efficient. 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. Figure 2 illustrates the ratio image traversal and depth edge detection.

We refer to our previous work [12] for more details on the algorithm and discussion with related methods. In spite of the robustness of our method in real-world scenes, it has limitations to detect multi-scale depth changes, and fails with object motion. Next we will describe these problems and provide solutions in a common multi-flash imaging framework.

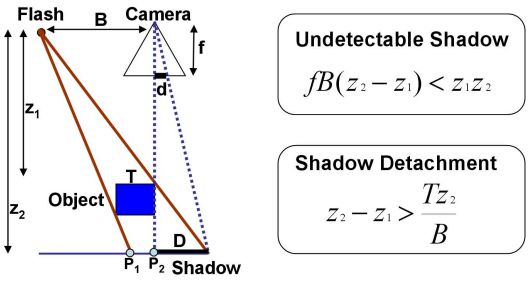
3. A Multi-Baseline Approach

Depth discontinuities in real world scenes are associated with different amounts of depth changes, referred as "jumps of discontinuities" by Birchfield [2]. Ideally, we want a method that is able to detect depth edges at different scales, ranging from tiny to large changes in depth. We will show how to deal with this problem by taking advantage of the spatial position of the light sources, in our multi-flash imaging framework.

3.1. Baseline Tradeoff

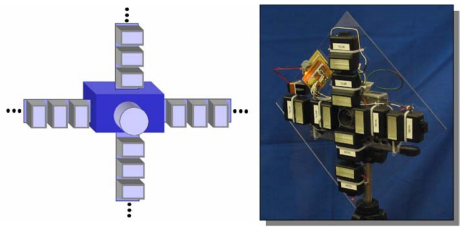
Depth edges associated with small changes in depth might be missed due to an undetectable narrow shadow width, caused by a small camera-flash baseline. On the other hand, a larger baseline may cause detached shadows (separated from the object), leading to false depth edges.

In order to analyze this problem in more detail, we look



(a) (b)

Figure 3: *(a) Relationship between baseline and shadow width. (b) Conditions for undetectable shadow and shadow detachment.*



(a) (b)

Figure 4: *(a) Our multi-baseline camera prototype. (b) Our implementation setup with two baseline levels.*

at the imaging geometry of the shadows, depicted in Figure 3a. The variables involved are f (camera focal length), B (camera-flash baseline), z_1, z_2 (depths to the shadowing and shadowed edges), D (shadow width) and d (the shadow width in the image plane). We have that $\frac{d}{f} = \frac{D}{z_2}$ and $\frac{D}{z_2 - z_1} = \frac{B}{z_1}$. It follows that the shadow width in the image can be computed as $d = \frac{fB(z_2 - z_1)}{z_1 z_2}$. For small depth changes in the scene, far away from the camera, it is possible that $fB(z_2 - z_1) < z_1 z_2$. In this case, the shadow will not appear in the image, leading to missing depth edges.

We note that there are two ways of solving this problem: either we improve camera resolution, with larger focal length f, or we use a wider camera-flash baseline B, which is more convenient, since camera resolutions are limited. However, a larger baseline may cause detached shadows, mainly in narrow objects. Let T be the width of the object. Analyzing Figure 3a, we note that as we increase the baseline, point P_1 moves towards point P_2. Shadow detachment will occur when point P_1 passes over point P_2. When they are at the same position, from the imaging geometry, we have that $\frac{T}{z_2 - z_1} = \frac{B}{z_2}$. It follows that if the amount of depth change $z_2 - z_1 > \frac{Tz_2}{B}$, the shadow will be separated from the object and a false depth edge will be marked. Figure 3b illustrates the two main conditions of the baseline tradeoff.

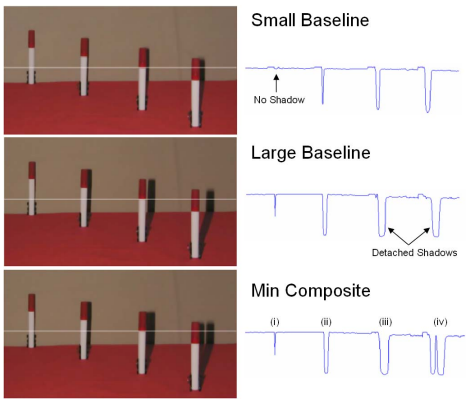


Figure 5: *Analysis of the four cases related to the baseline tradeoff, considering a narrow object.*

3.2. Proposed Solution

Our approach to handle the baseline tradeoff is to use a multi-baseline photometric method, where extra light sources are placed at different baselines, as shown in Figure 4a. With this new configuration, we are able to detect depth edges associated with small changes in depth (using large baselines), without creating shadow separation in narrow objects (using small baselines).

The main question is how to combine the information of the images taken with flashes at different baselines. A simple, yet effective way is to just take the minimum composite among the images. Provided that the light sources are sufficiently close to each other, the shadows in different baselines will merge, avoiding detached shadows, while preserving sufficiently large shadow widths.

In our implementation setup, we used two different baselines (see Figure 4b). Let F_S and F_L be the small and large baseline flashes, respectively. There are essentially four cases we need to consider at depth edges (Figure 5):(i) F_S creates an undetectable narrow shadow and F_L creates a detectable shadow; (ii) F_S creates a detectable small width shadow and F_L creates a larger width shadow; (iii) F_S creates a detectable shadow but F_L creates a detached shadow that overlaps with F_S shadow; (iv) same as (iii) but the shadows of F_S and F_L do not overlap.

Note that in the first three cases, the minimum composite solution is suitable, but in the fourth case, there is a non-shadowed region between the shadows created by F_S and F_L. This would lead to a false depth edge along the shadow created by F_L. Next we describe an algorithm to handle this problem.

3.2.1 Eliminating Detached Shadows

Our algorithm is based on the observation that if the start point of a shadow created by F_S is not the start point of a shadow created by F_L, then the next shadow along the epipolar ray created by F_L is a detached shadow. Figure 6

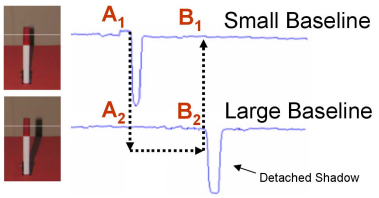


Figure 6: *Algorithm for eliminating detached shadows when the light sources are not sufficiently close to each other.*

shows the values of the *ratio images* along a scanline for F_S and F_L, when shadow detachment occurs. Our algorithm can be summarized in the following steps:

- Traverse along the epipolar rays the ratio images R_S and R_L, associated with F_S and F_L, respectively.
- If a depth edge appears in R_S and not in R_L (see points A_1 and A_2 in Figure 6) :
  - Traverse R_L along the epipolar ray until the next detected depth edge (see point B_2 in Figure 6).
  - If at this position there is no correspondent depth edge in R_S (see point B_1 in Figure 6), we mark this edge as a spurious, detached shadow edge.

The last step is important to confirm the presence of a detached shadow as well as to ensure that no edges detected with the small baseline flash F_S will be marked as spurious.

Note that using this algorithm eliminates the problem in case (iv), when shadows of F_S and F_L do not overlap. This solution will fail to detect depth discontinuities when even F_L does not create a detectable shadow or for very thin objects, where even F_S creates a detached shadow. In this case, extra light sources could be added to our setup.

3.3. Multi-Scale Depth Edges in Real-World Scenes

Figure 7a shows an image captured with a small camera-flash baseline. Since there are very small depth changes in the interior region of the pinecone, depth edges are missed in this region (Figure 7b). On the other hand, if we use a large baseline, the shadow gets detached from the basket (Figure 7c), leading to a false depth edge (Figure 7d). Using our multibaseline approach with two levels of baseline, we are able to eliminate detached shadows, and still preserve depth edges associated with small changes in depth, as shown in Figure 7e.

A more complex example is depicted in Figure 8a, which shows the image of a car engine, containing depth edges associated with different amounts of depth changes. Figure 8b shows intensity edge detection for this image, using the Canny operator. Note that important shape boundaries are missing due to low contrast variations, while edges due to texture variations not associated with occluding edges confound scene structure.

We compared our multibaseline approach with the naive single-baseline algorithm in such complex scene (Figures 8c and 8d). Note that our method captures depth edges associated with tiny and larger changes in depth, which is not possible with our previous setup. Figure 8e better illustrates this comparison, zooming into a specific part of the engine.

In our implementation setup, the baselines are about 50mm and 100mm. Using a 4.0 MegaPixel Canon G3, we verified that we can capture depth discontinuities with changes in depth as small as 5mm at a distance no larger than 2000mm from the camera. We can handle large depth changes in room-sized environments, unless the object is sufficiently narrow to cause shadow detachment with the small baseline flash. In this case, extra light sources could be placed at smaller baselines to handle this problem. We do not address here the case where no shadows are created due to a distant background, although we have discussed this issue in our previous work [12].

4 Variable Wavelength

So far, our method requires taking multiple pictures of the same static scene. This clearly poses a problem for detecting depth edges in motion. Due to the lack of simultaneity, the base maximum composite image will have misaligned features, leading to spurious edges.

In order to handle this problem, we could use a high speed camera, with flashes triggered in a rapid cyclic sequence, syncronized with the camera video frames. We note that a high speed camera can reduce the amount of motion between frames, but still the frame simultaneity cannot be assumed. A reasonable approach is to apply motion compensation techniques to correct the introduced artifacts. Finding optical flow and motion boundaries, however, is a challenging problem, mainly in textureless regions [2].

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. Our approach is part of our common multi-flash framework, relying on the variation of the wavelength of the light sources. Very little attention has been given to photometric vision methods that make use of flat colored lights. We demonstrate here that by using light sources with different colors, we can trigger them all at the same time, in one single shot, and then exploit the colored shadows to extract depth edges.

Figure 9a shows our setup with three lights of different color: red, green and blue. When the lights are triggered at the same time, they sum to white light. We appropriately

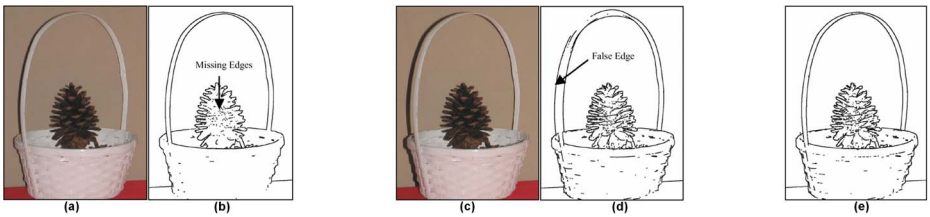


Figure 7: *(a) Small baseline image and (b) correspondent depth edges. (c) Large baseline image with shadow detachment and (d) correspondent depth edges. (e) Our final result using our multibaseline approach.*

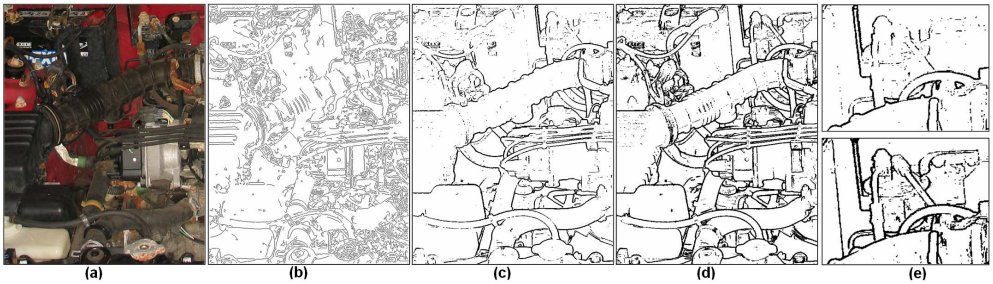


Figure 8: *(a) Complex scene with different amounts of changes in depth. (b) Canny edge detection. (c) Depth edges computed with one single camera-flash baseline. (d) Depth Edges computed with our multibaseline approach. (e) Part of the engine zoomed in to compare single-baseline depth edges (top) with our multibaseline method (bottom).*

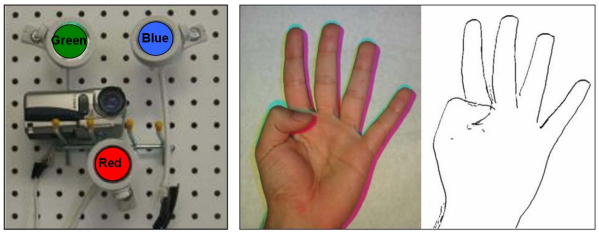


Figure 9: *(a) Our setup for dynamic scenes with different wavelength light sources. (b) Input image. Note the shadows with different colors. (c) Depth edge detection.*

placed the lights to produce shadows along all depth discontinuities in the scene: one is positioned below the camera, while the others are placed on the left and right upper diagonals of the camera center of projection.

Our algorithm to detect depth edges using this setup follows a similar idea of the algorithm described in section 2. Given the input image with colored shadows, we need to first distinguish which shadows were created by which light source. With this information, we can traverse the image along the correspondent epipolar rays, marking depth edges at shadows associated with correspondent light sources. In our setup, for the lights placed along the camera diagonal, the traversal is not aligned with the pixel grid. For efficiency, we may keep the traversal along the pixel grid, but detecting negative transitions with e.g., steerable kernels tuned to specific directions.

A simple way to distinguish the shadows created by each light would be to decompose the input image into the red, green and blue channels. However, finding shadows in each channel is not an easy problem. In this case, the maximum composite image of the three channels is not suitable for computing a shadow-free image, due to the fact that non-shadowed regions of each channel may have different intensities, depending on the albedo of the objects.

In the following section we will describe an approach to segment shadows using a reference image of the scene, captured with white light sources. This method does not solve the motion problem, but considerably reduces acquisition time, allowing motion compensation algorithms to work better. Then, in section 4.2, we rely on shadow detection from a single image to extract depth edges in dynamic scenes. Although separating shadow edges from reflectance edges is a difficult problem for general scenes, we show that our technique can be useful for specific applications, such as extracting internal hand contours or lip segmentation from video.

4.1 Using a Reference Image

The basic idea of our method based on a reference image is to take two pictures of the scene, one with the three red, green, and blue lights triggered at the same time, and the other with white light sources. At non-shadowed regions, the ratio between the images should be close to one, due to the fact that red, green and blue sum to white light. The ratio in each channel can be used to distinguish which shadows

were created by which light source.

We now describe this idea in more detail. First consider the RGB color ρ_k, $k = R, G, B$ formed at a particular pixel, for illumination with spectral power distribution $E(\lambda)$ impinging on a surface with spectral reflectance function $S(\lambda)$. If the three camera sensor sensitivity functions form a set $Q_k(\lambda)$, $k = R, G, B$, then we have:

$$\rho_k = \sigma \int E(\lambda)S(\lambda)Q_k(\lambda)\, d\lambda, \quad k = R, G, B \qquad (1)$$

where σ is Lambertian shading, i.e. the inner product between lighting direction and surface normal at a particular surface point. Assuming that camera sensitivities are Dirac delta functions, $Q_k(\lambda) = q_k\delta(\lambda - \lambda_k)$, then equation 1 is simplified:

$$\rho_k = \sigma E(\lambda_k)S(\lambda_k), \quad k = R, G, B \qquad (2)$$

The capture process of our technique consists in first taking an image I_{color} of the scene with three light sources red, green, and blue triggered at the same time. Then, we replace the colored lights with white lights of same intensity and capture a reference image I_{white}. The white light sources could be placed near the colored lights (to avoid replacing the lights), provided that the scene depth is sufficiently large when compared to the baseline between camera and lights. We assume the two images are properly registered or the scene is static between the shots. Note that we keep all imaging parameters constant, except the spectral power distribution of the light sources. By taking the ratio between each channel of I_{color} and I_{white}, we have:

$$S_k = \frac{I_{color_k}}{I_{white_k}} = \frac{\sigma E_{color}(\lambda_k)S(\lambda_k)q_k}{\sigma E_{white}(\lambda_k)S(\lambda_k)q_k} = \frac{E_{color}(\lambda_k)}{E_{white}(\lambda_k)}, \quad k = R, G, B \qquad (3)$$

where E_{color} and E_{white} correspond to the combined spectral distribution of the three colored and white light sources, respectively. As we can see from the equation above, the reflectance term is canceled, which is important for detecting shadows without depending on the albedo of objects in the scene.

Given that the light sources may have different intensities, we convert I_{color} and I_{white} to a chromatic space before taking the ratio in equation 3. More specifically, we define $I'_{color_k} = \frac{I_{color_k}}{\sum_{i=R,G,B} I_{color_i}}$ and $I'_{white_k} = \frac{I_{white_k}}{\sum_{i=R,G,B} I_{white_i}}$ for $k = R, G, B$. These intensity normalized images are shown in Figures 10b and 10c. The ratio $\frac{I'_{color}}{I'_{white}}$ is shown in Figure 10d. Ratio values greater than one are clamped to one. Note that shadows can be easily segmented in this image for depth edge detection. The reason is that at non-shadowed regions, the ratio between the images is close to one, due to the fact that red, green and blue sum

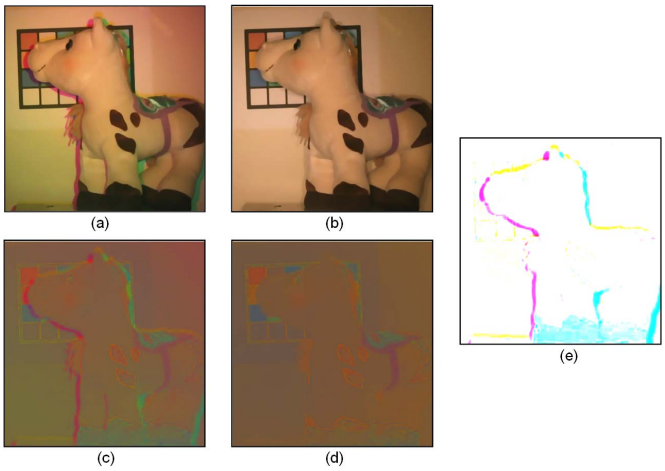


Figure 10: *(a) Image I_{color} taken with red, green and blue light sources. (b) Image I_{white} taken with white light sources. (c) Conversion to chromatic space: I'_{color} (d) I'_{white} (e) ratio between I'_{color} and I'_{white}. The color of the segmented shadows indicates which light source corresponds to each shadow.*

to white light and thus $E_{color} \approx E_{white}$. On the other hand, at shadowed regions, at least one of the light sources does not illuminate the local region, implying a drop in the ratio image due to the different spectral distribution of E_{color} and E_{white}.

For red, green and blue lights, we used 50W Ushio Popstar MR-16 Halogen light bulbs, with 12° beam angle spread. For white light sources, we used three 50W Ushio Whitestar MR-16 Halogen light bulbs, also with 12° beam angle spread. The presence of some artifacts in Figure 10d are mostly due to the narrow beam angle spread of light sources. Since the lights are not precisely calibrated, parts of the scene may have more incidence of light sources at specific wavelengths. This may create spurious transitions on the ratio image, leading to false shadow segmentation (such as along the horse foot).

4.2 Learning Shadow Color Transitions

We now turn to the problem of detecting depth edges with a single image, captured with red, green and blue light sources triggered at the same time, as shown in Figure 9a. In this case, detection of depth edges can be achieved by using algorithms that detect shadows [6] or separate illumination from reflectance using a single image [17].

Finlayson et al. [6] proposed a method to remove shadows from images by deriving a 1D illumination invariant image representation based on log-chromaticity coordinates. Tappen et al. [17] use color information and a classi-

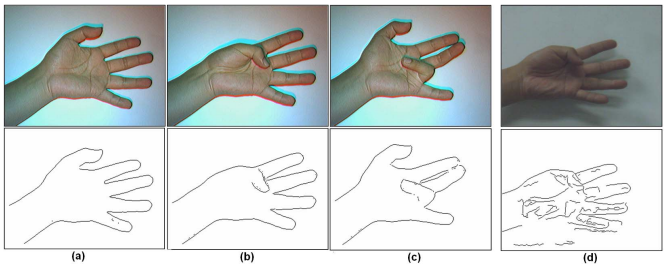


Figure 11: *(a)-(c) Sample frames of a video sequence and correspondent depth edge detection. (d) Comparison with Canny edge detection.*

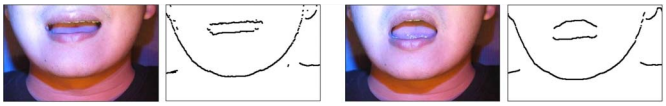


Figure 12: *Lip contour extraction using two red and blue lights placed above and below the camera.*

fier trained to recognize gray-scale patterns in order to classify image derivatives as being caused by reflectance or illumination changes. The retinex algorithm [9] addresses the same problem, but relying on the assumption that the gradients along reflectance edges have larger magnitude compared to those caused by illumination variation.

Although these algorithms could be exploited to segment shadows in general scenes, we have implemented a simpler method that relies on learning the color transitions between object and shadows. This allows us to distinguish which light source created a specific shadow. More specifically, in a training stage we collect sample pixels along the shadow transitions and project a support vector machine classifier for each light source. During the epipolar ray traversal, we use the output of the correspondent classifier to mark depth edges. Anisotropic diffusion [11] was applied as preprocessing for noise filtering.

Our method may generate false positives along reflectance edges with similar learned color transitions. Depending on the background albedo, shadow color transitions may differ from the learned model, thus leading to false negatives. Despite these limitations, our technique can be useful for diferent applications. For example, extracting depth edges due to finger occlusions is extremely important for hand gesture analysis and recognition. In fact, we have recently demonstrated that knowlege about occluding edges in the hand significantly improves recognition rate over standard intensity edges [5]. Other examples include extraction of lip contours and interior edges of the ear for recognition.

Figures 9b and c shows an input image and our final result, respectively. Note that we are able to capture the self-occluding thumb finger edge, while eliminating texture edges such as wrinkles in the hand. This would not be possible with intensity edge detectors. Although background clutter could be a source of noise for our method, the hand could be extracted using standard segmentation techniques (e.g., based on skin color or background subtraction), while applying our technique just to extract interior occluding contours.

When video sequences are available, we use space-time consistency to improve depth edge detection. We basically consider the depth edge frames as a 3D surface, filling out edge gaps among frames to ensure surface smoothness. Figure 11 shows sample frames of a video sequence captured with our light sources with different wavelength. We compare our results with intensity edges detected with the Canny operator. Note that Canny edges include undesirable texture edges, such as wrinkles and nails, while missing important self-occluding edges due to low intensity variations.

Another example is shown in Figure 12, using only two red and blue lights, placed below and above the camera, respectively. Note that we are able to reliably detect the upper and lower lip contours, while reducing noise inherent in intensity edge detection. Lip contour extraction would not be possible if the mouth is closed, but still detecting whether the mouth is open or not could be useful for speech analysis or facial expression recognition. Only two lights are sufficient for this example, since lip contour edges are mostly horizontal. We also extract the facial contour, which is a challenging task for intensity edge detectors due to the low contrast variation between the face and neck area.

4.3 Discussion

In our previous work [12], we have addressed the problem of detecting depth edges in motion using light sources triggered in a rapid cyclic sequence, syncronized with the camera video frames. However, this method assumes that motion is monotonic with small magnitude, which may cause spurious edges for thin structures or objects with high frequency texture in the scene.

Most approaches that use active colored illumination in computer vision aim to recover the full 3D information of the scene through structured lighting. Tajima and Iwakawa [14]use a rainbow pattern for pixel coding and triangulation. Sa et al. [13] adopt color coding along projected colored stripe boundaries in time, which is useful for 3D reconstruction from dynamic scenes. Compared to these techniques, our approach offers the advantage of being simple, inexpensive, and easily built into a self-contained device. It could also complement existing stereo techniques to obtain depth edge preserving 3D reconstruction.

4.3.1 Limitations

Our method is not suitable for distant objects or outdoor scenes, where the intensity of the light sources may be insufficient compared to the sun light. Objects very close to the camera may suffer from pixel saturation, also violating the assumption that the scene depth is significantly larger than the camera-light baseline. Another problem occurs when the color of the shadow is the same as the background. For example, if a background pixel has color yellow and is not illuminated by the blue light source, an yellow shadow (formed by the combination of red and green lights) would not be detectable.

Our technique based on a single shot capture using lights with different wavelength (section 4.2) is limited to handle general scenes, as false negatives and positives may arise due to the albedo of objects in the scene. Research on shadow segmentation and intrinsic image computation from a single frame [17, 6] is very important to achieve a more general solution to the problem. We also believe that solutions involving new camera setups would be possible. For example, infra-red lighting could be used (with different wavelength light sources) to capture an image with shadows, while using another camera at the same viewpoint (this is possible using a beamsplitter) to capture an ambient image simultaneously. This ambient image could be used as reference to segment shadows more reliably.

5. Conclusions

In this paper, we demonstrated that we are able to handle fundamental problems in depth edge detection - including multi-scale depth changes and motion - in a common framework, based on the variation of illumination parameters, such as the number, spatial position, and wavelength of the light sources.

As future work, we plan to analyze other illumination parameters (such as controlling the intensity/power and duration of the flashes, or using linear light sources) to improve depth edge detection. We also intend to do a more systematic evaluation and extend our multi-flash imaging framework to handle the general problem of classification of discontinuities according to their physical origin, i.e, discriminating discontinuities in depth, reflectance, illumination and surface normal.

References

[1] M. Agrawal and L. Davis. Window-based, discontinuity preserving stereo. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.

[2] S. Birchfield. *Depth and Motion Discontinuities*. PhD thesis, Stanford University, 1999.

[3] R. Feris, R. Raskar, L. Chen, K. Tan, and M. Turk. Discontinuity preserving stereo with small baseline multi-flash illumination. In *International Conference on Computer Vision (ICCV'05)*, Beijing, China, 2005.

[4] R. Feris, R. Raskar, k. Tan, and M. Turk. Specular reflection reduction with multi-flash imaging. In *IEEE Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'04)*, Curitiba, Brazil, 2004.

[5] R. Feris, M. Turk, R. Raskar, K. Tan, and G. Ohashi. Exploiting Depth Discontinuities for Vision-based Fingerspelling Recognition. In *IEEE Workshop on Real-time Vision for Human-Computer Interaction (in conjunction with CVPR'04)*, Washington DC, USA, 2004.

[6] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006.

[7] E. Gamble, D. Geiger, T. Poggio, and D. Weinshall. Integration of vision modules and labeling of surface discontinuities. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1576–1581, 1989.

[8] S. Gokturk and C. Tomasi. 3D head tracking based on recognition and interpolation using a time-of-flight depth sensor. In *Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004.

[9] E. Land and J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61:1–11, 1971.

[10] D. Marr. *Vision: a Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, 1982.

[11] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[12] R. Raskar, K. Tan, R. Feris, J. Yu, and M. Turk. A nonphotorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *SIGGRAPH'04 / ACM Transactions on Graphics*, 2004.

[13] A. Sa, P. Carvalho, and L. Velho. (b, s)-bcsl : Structured light color boundary coding for 3d photography. In *International Fall Workshop on Vision, Modeling, and Visualization*, 2002.

[14] J. Tajima and M. Iwakawa. 3d data acquisition by rainbow range finder. In *International Conference on Pattern Recognition*, pages 309–313, 1990.

[15] K. Tan, J. Kobler, P. Dietz, R. Feris, and R. Raskar. Shape-Enhanced Surgical Visualizations and Medical Illustrations with Multi-Flash Imaging. In *International Conference on Medical Imaging Computing and Computer Assisted Intervention (MICCAI'04)*, France, 2004.

[16] M. Tappen and W. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *International Conference on Computer Vision*, Nice, France, 2003.

[17] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1459–1472, 2005.