

Mobile Multi-flash Photography

Xinqing Guo^a, Jin Sun^b, Zhan Yu^a, Haibin Ling^c, Jingyi Yu^a

^aUniversity of Delaware, Newark, DE 19716, USA;

^bUniversity of Maryland, College Park, MD 20742, USA;

^cTemple University, Philadelphia, PA 19122, USA;

ABSTRACT

Multi-flash (MF) photography offers a number of advantages over regular photography including removing the effects of illumination, color and texture as well as highlighting occlusion contours. Implementing MF photography on mobile devices, however, is challenging due to their restricted form factors, limited synchronization capabilities, low computational power and limited interface connectivity. In this paper, we present a novel mobile MF technique that overcomes these limitations and achieves comparable performance as conventional MF. We first construct a mobile flash ring using four LED lights and design a special mobile flash-camera synchronization unit. The mobile device's own flash first triggers the flash ring via an auxiliary photocell. The mobile flashes are then triggered consecutively in sync with the mobile camera's frame rate, to guarantee that each image is captured with only one LED flash on. To process the acquired MF images, we further develop a class of fast mobile image processing techniques for image registration, depth edge extraction, and edge-preserving smoothing. We demonstrate our mobile MF on a number of mobile imaging applications, including occlusion detection, image thumbnailing, image abstraction and object category classification.

Keywords: Multi-flash Camera, Non-photorealistic Rendering, Occluding Contour

1. INTRODUCTION

Multi-flash (MF) photography takes successive photos of a scene, each with a different flashlight located close to the camera's center of projection (CoP). Due to the small baseline between the camera CoP and the flash, a narrow sliver of shadow would appear attached to each depth edge. By analyzing shadow variations across different flashes, we can robustly distinguish depth edges from material edges.¹ MF photography hence can be used to remove the effects of illumination, color and texture in images as well as highlight occluding contours. Previous MF cameras, however, tend to be bulky and unwieldy in order to accommodate the flash array and the control unit. In this paper, we present a mobile MF photography technique suitable for personal devices such as smart phones or tablets.

Implementing mobile MF photography is challenging due to restricted form factor, limited synchronization capabilities, low computational power and limited interface connectivity of mobile devices. We resolve these issues by developing an effective and inexpensive pseudo flash-camera synchronization unit as well as a class of tailored image processing algorithms. We first construct a mobile flash ring using four LED lights and control it using the mobile device's own flash. Specifically, the mobile flash first triggers the flash ring via an auxiliary photocell, as shown in Fig. 1. It then activates a simple micro-controller to consecutively trigger the LED flashes in sync with the mobile camera's frame rate, to guarantee that each image is captured with only one LED flash on.

To process the acquired MF images, we further develop a class of fast mobile image processing techniques for image registration, depth edge extraction, and edge-preserving smoothing. We demonstrate our mobile MF on a number of mobile imaging applications, including occlusion detection, image thumbnailing, and image abstraction. We also explore the depth-edge-assisted category classification based on mobile MF camera. Compared with traditional MF cameras, our design is low-cost (less than \$25) and compact ($1.75'' \times 2.75''$). Our solution is also universal, i.e., it uses the device's flash, a universal feature on most mobile devices, rather than device-specific external interfaces such as USBs. Experimental results show that our mobile MF technique is robust and efficient and can benefit a broad range of mobile imaging tasks.

Further author information:

Xinqing Guo: E-mail: xinqing@udel.edu, Telephone: +1(302)561-0292

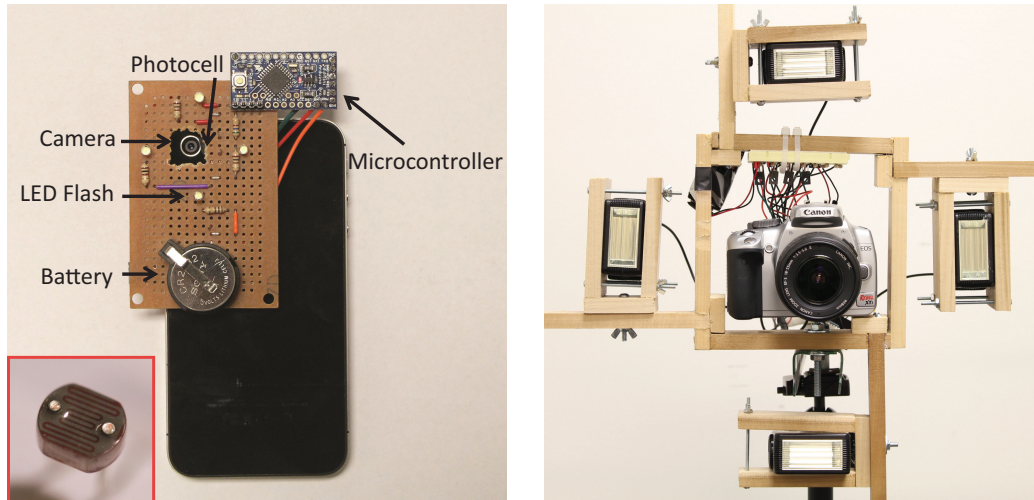


Figure 1. (Left) Our prototype mobile MF system. The photocell is hidden on the back of the system. The red highlighted region shows the closeup of the photocell. (Right) Traditional MF system with SLR-camera.

2. RELATED WORK

Flash-based computational photography has attracted much attention in the past decade. Earlier approaches aim to enhance imaging quality by fusing photographs captured with and without flash. The seminal flash/no-flash pair imaging applies edge preserving filters to enhance noisy no-flash images with high quality flash images. Eisemann and Durand² and Petschnigg et al.³ used the no-flash image to preserve the original ambient illumination while inserting sharpness and details from the flash image. Krishnan et al.⁴ explored the use of non-visible light (UV/IR) flashes and demonstrated how different wavelength imagery can be used to for image denoising.

Raskar et al. presented the first multi-flash camera¹ that uses an array of flashes surrounding the central SLR camera. They take multiple shots of the scene, each with only one flash. Each flash casts a different shadow abutting the occlusion boundary of the object and they extract the boundaries by traversing along the flash-camera epipolar line. Feris et al.⁵ further show that one can conduct stereo matching using MF photography. They derived object depths (disparities) in terms of shadow widths and then applied belief propagation for scene reconstruction. Liu et al.⁶ mounted MF cameras on robots for enhancing object detection, localization and pose estimation in heavy clutter.

Previous MF photography is sensitive to specular surfaces, thin objects, lack of background, and moving objects, and a number of extensions have been proposed to address these issues. To find proper flash-camera configuration, Vaquero et al.⁷ investigated the epipolar geometry of all possible camera-light pairs to characterize the space of shadows. Their analysis can be used to derive the lower bound on the number of flashes, as well as the optimal flash positions. Tremendous efforts have also been made to reduce the number of flashes or shots in MF. Feris et al.⁸ used color multiplexing to more robustly handle multi-scale depth changes and object motion. They have shown that for some special scene configurations, a single shot with the color flash is sufficient for depth edge extraction whereas for general scenes, a color/monochrome flash pair would be enough. Most recently Taguchi et al.⁹ utilized a ring color flashes of continuous hues for extracting the orientation of depth edges.

MF photography has broad applications. On the computer graphics front, the acquired depth edges can be used to synthesize non-photorealistic effects such as line-arts illustrations,¹⁰ image abstraction,¹ and image thumbnailing.¹¹ On the computer vision front, recent studies have shown that the depth edges can significantly improve visual tracking and object recognition.¹² We explore these applications on mobile devices by developing efficient image processing schemes.

Finally, our work is related to emerging research on mobile computational photography. Mobile devices have seen exponential growth in the past decade. For example, the latest iPhone 5S features a 1.3 GHz dual core 64 bit CPU, 1 GB RAM and an 8 megapixel camera. Samsung Galaxy S4 features a 1.9 GHz quad core, 2 GB

RAM and a camera with similar quality as iPhone 5S. Numerous efforts have been made to migrate conventional computational photography algorithms such as high-dynamic-range imaging,¹³ panorama synthesis,¹⁴ light field rendering,¹⁵ etc., onto mobile platforms. The latest effort is the FCam API by Adams et al.¹⁶ to allow flexible camera controls. For example, Nvidia's Tegra 3, the world's first quad-core full-featured tablet, directly uses FCam for controlling its 5 megapixel stereo camera with flash. Our paper explores a related but different problem of controlling flashes on mobile platform, where the FCam API is not directly applicable.

3. MOBILE MULTI-FLASH HARDWARE

3.1 Construction

Figure 1 shows our prototype mobile MF device that uses a micro-controller to trigger an array of LED flashes. To control the micro-controller, the simplest approach would be to directly use the mobile device's external interface, e.g., the USB. For example, the recent Belkin camera add-on for iPhone allows user to have a more camera-like hold on their phone while capturing images by connecting to the data port. However, this scheme has several disadvantages. First, it requires additional wiring on top of the already complex setup. Second, it will occupy the USB interface and limit the use of other application. Finally, each platform (Samsung vs. Apple vs. Nokia) will need to implement its own version of the control due to heterogeneity of the interface. Other alternatives include the Wi-Fi and the audio jack. However, it would require modifying sophisticated circuitry and the communication protocols.

Our strategy is to implement a cross-platform solution: we use the original flash on the mobile device to trigger the LED flashes. We implement our solution on a perfboard. To reduce the form factor, we choose the Arduino pro mini micro-controller, a minimal design approach ($0.7'' \times 1.3''$) of the Arduino family. We also use small sized but bright LEDs, e.g., the 3mm InGaN white LED from Dialight with a luminous intensity of 1100 mcd and a beam angle of 45 degree. It is worth noting that brighter LEDs are available but many require higher forward current which can cause damage to the micro-controller. In our setup, the baseline between the LED and the camera is about $0.6''$.

To trigger the micro-controller, we put a photocell in front of the device's own flash. The photocell serves as a light sensor that takes the flash signal from the mobile device to trigger the multi-flash array. In our setup, we use a CdS photoconductive photocell from Advanced Photonix. The photocell is designed to sense light from 400 to 700 nm wavelength and its response time is around 30 ms. The resistance is 200k Ohms in a dark environment and will drop to 10k Ohms if illuminated at 10 lux. The complete system is powered by two button cell batteries, making it self-contained. Its overall size is $1.75'' \times 2.75''$ and therefore can be mounted on a wide range of mobile devices, ranging from the iPhone family to the Samsung Galaxy and Note family. For example, even for the smallest sized iPhone 4/4S ($2.31'' \times 4.54''$), our system fits perfectly.

3.2 Image Acquisition

To avoid the device's flash to interfere with the LED flashes, we initiate the image acquisition process only after the device's flash goes off. The frame rates of the camera and the LED flash ring are set to be identical by software (e.g., the AVFoundation SDK for iOS) and by micro-controller respectively. After acquiring four images, we turn on the device's flash to stop the acquisition module. We also provide a quick preview mode to allow users to easy navigate the captured four images. If the user is unsatisfied with the results, with a single click, he/she can reacquire the image and discard the previous results.

Conceptually, it is ideal to capture images at the highest possible frame rate of the device (e.g., 30 fps on iPhone 4S). In practice, we discover that a frame rate higher than 10 will cause the camera out-of-sync with the flash. This is because the iPhone and the Arduino micro-controller use different system clocks and are only perfectly sync'd at the acquisition initiation stage. In our implementation, we generally capture four flash images at a resolution of 640×480 images in 0.4s. The low frame rate can lead to image misalignment since the device is commonly held by a hand. We compensate for hand motion by applying image registration (Section 4.1) directly on mobile devices.

A unique feature of our system is its extensibility, i.e., we can potentially use many more flashes if needed. The Arduino pro mini microcontroller in our system has 14 digital I/O pins: one serves as an input for the

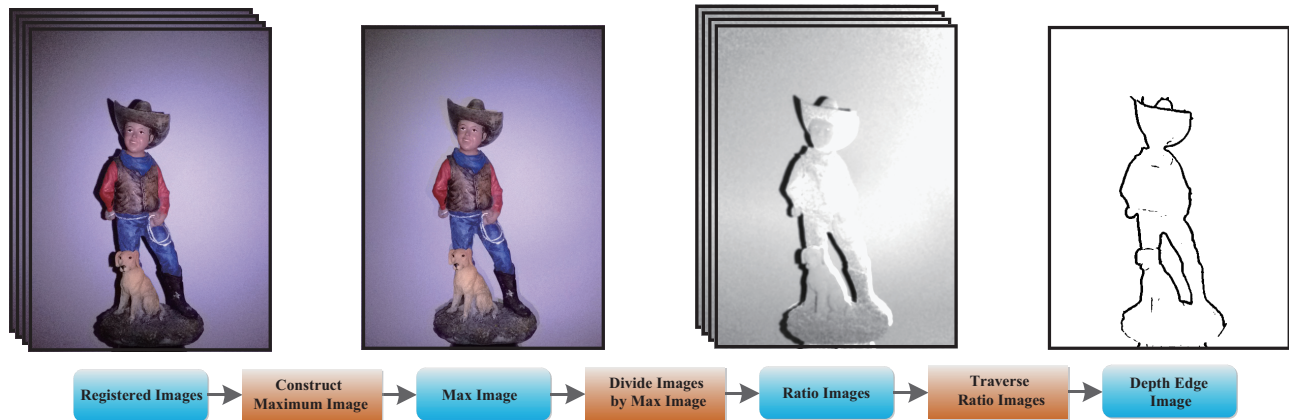


Figure 2. The pipeline for depth edge extraction.

triggering signal and the others as output for the LED flashes. Therefore, in theory, we can control 13 flashes with minimum modification.

4. MF IMAGE PROCESSING

4.1 Depth Edge Extraction

Traditional MF photography assume that the images are captured from a fixed viewpoint. In contrast, our mobile MF photography uses a hand-held device and the images are usually shifted across different flashes as we capture with a low frame rate. Extracting depth edges without image alignment will lead to errors as shown in Fig. 4(b). In particular, the texture edges are likely to be detected as depth edges. We therefore implement a simple image registration algorithm by first detecting SIFT features and then use them to estimate the homography between images. This scheme works well for scenes that contain textured foreground (Fig. 6) or background (Fig. 5). It fails in the rare scenario that the scene contains very few textures and the shadow edges become the dominating SIFT features in homography estimations.

Once we align the images, we adopt the shadow traversing algorithm in¹ to extract the depth edges. Figure 2 shows the processing pipeline. The captured MF images contain noise, especially under low-light environment. We therefore first convert the color images to grey scale and apply Gaussian smoothing. We denote the resulting four images as $I_k, k = 1..4$ and construct a maximum composite image I_{max} where $I_{max}(x, y) = \max_k(I_k(x, y))$. To detect the shadow regions, we take the ratio of a shadow image with the maximum composite image as $R_k = I_k / I_{max}$. The ratio is close to 1 for non-shadow pixels and is close to 0 for shadow pixels. A pixel on the depth edge must transition from the non-shadow region to the shadow region and we apply Sobel filter on each of the ratio images to detect such transitions. In the final step, we apply a median filter to the depth edge image to further suppress the noise. The complete process takes about 1.2s for images with a resolution of 640×480 on an iPhone 4S.

4.2 Non-photorealistic Rendering

From the depth edge image, we can further perform post-processing techniques for synthesizing various non-photorealistic effects.

Line-art Rendering. Line-art image is a simple yet powerful way to display an object. Lines not only represent the contour of an object but also exhibit high artistic value. Raskar et al.¹ convert the edge image to a linked list of pixels via skeletonization and then re-render each edge stroke. However, it is computationally expensive. We adopt a much simpler approach using simple filtering. We first downsample the image by bicubic interpolation, then apply the gaussian filter, and finally upsample the image. Both bicubic interpolation and gaussian filter serve as low pass filters, which will blur the binary depth edge image. Also users are capable of adjusting the



Figure 3. (Left) Image abstraction by using anisotropic diffusion. (Right) Image abstraction by using bilateral filter.

kernel size to control the smoothness. Our processing pipeline is simple, making it suitable for implementation on the mobile platform. iPhone 4S takes about half a second for processing an 640×480 image.

Image Abstraction. The most straightforward approach is to use edge-preserving filters such as bilateral filters or anisotropic diffusion¹⁷ to suppress texture edges while preserving depth edges. For example, we can apply the joint bilateral filters³ that uses the depth image for computing the blur kernel and then blurring the max image I_{max} . A downside of this approach is that the result may exhibit color blending across the occlusion boundaries, as shown in Fig. 3. This is because bilateral filters do not explicitly encode the boundary constraint in the blurring process, i.e., the contents to the left and to the right of the edge are treated equally.

To avoid this issue, we apply anisotropic diffusion instead. Specifically, we diffuse the value of each pixel to its neighboring pixels iteratively and use the depth edge as constraints. To ensure that pixels will not diffuse across the depth edge, at the n th iteration, we compute the mask M_n

$$M_n(x, y) = \begin{cases} I_n(x, y) & \text{if } (x, y) \notin \text{edge pixel} \\ 0 & \text{if } (x, y) \in \text{edge pixel} \end{cases}$$

and

$$I_{n+1}(x, y) = \frac{w \sum_{(x_t, y_t) \in N} M_n(x_t, y_t) + M_n(x, y)}{1 + 4w} \quad (1)$$

where N are the neighboring pixels to (x, y) and w is the assigned weight to the neighboring pixels. In our implementation, we simply set $w = 5$. Notice that large w will make the diffusion converge faster and we limit the number of iterations to 15. Finally, we add the edge map to the texture de-emphasized results. On an iPhone 4S, this process takes about 1.5s.

Image Thumbnailing. Image thumbnailing reduces the size of the normal image for better organizing and storing. By using bicubic interpolation, we can downsample the texture de-emphasized image to create a stylized thumbnail image. The depth edges are preserved while the texture regions are blurred, making it suitable for creating icons.

5. OBJECT CATEGORY CLASSIFICATION USING DEPTH EDGES

The effectiveness of using depth edges (occluding contours) in object category classification has been reported by recent study.¹² Specifically, depth edges can serve as *feature filter* which help high-level vision tasks to get

“purified” shape related features. Here we use similar bag-of-visual-word classification framework as in¹² for evaluation on a dataset collected by the proposed mobile multi-flash camera.

Category Classification Using Bag-of-Visual-Word Model. The main idea of bag-of-visual-word (BOW) approach is to represent image as histogram of visual words. 128-dimensional SIFT descriptor is used as independent feature. The dictionary of visual words is learned from training data using clustering method such as k-means. Each training and testing image is represented by histogram of visual words in the dictionary. A classifier is then learned in the space of these visual words for classification task. In this experiment we use Support Vector Machine (SVM) due to its simplicity and discriminative power. As for implementation detail, we chose the LibSVM package and Gaussian kernel.

Feature Filtering Using Depth Edges. Sun et al.¹² proposed to enhance the BOW framework by filtering out irrelevant features in images using depth edges. Let an image be $I : \Lambda \rightarrow [0, 1]$, where $\Lambda \in R^2$ defines the 2D grid. The set of feature descriptors are:

$$\mathcal{F}(I) = \{(\mathbf{x}_i, \mathbf{f}_i)\}, \quad (2)$$

where \mathbf{x}_i is the position of the i^{th} feature \mathbf{f}_i . After obtaining the depth edge image I_{DE} according to steps mentioned in previous sections, any feature that is far away from valid nonzero I_{DE} pixels will be eliminated. The new feature set \mathcal{G} is defined as:

$$\mathcal{G}(\mathcal{F}(I), I_{mask}) = \{(\mathbf{x}_i, \mathbf{f}_i) \in \mathcal{F}(I) \mid I_{mask}(\mathbf{x}_i) < \tau\}, \quad (3)$$

where $I_{mask}(\cdot)$ is the distance transform map of $I_{DE}(\cdot)$ and τ is a preset distance threshold. After filtering, feature descriptors become concentrated around depth edges of objects.

6. IMPLEMENTATION AND APPLICATION

6.1 Implementation

We have implemented our mobile MF system on an iPhone 4S. iPhone 4S features a 1 GHz dual core, a 512 MB RAM and an 8 megapixel camera with a fixed aperture of f/2.4. All examples in this paper are captured and rendered at an image resolution of 640×480 . The images are captured under indoor conditions to avoid outdoor ambient light overshadowing the LED flash light which would make it difficult to identify shadow regions. Further, the iPhone 4S does not allow the user to control the shutter speed. As a result, under a relatively dim environment, the camera uses a high ISO setting and the acquired images, even under the LED flashes, exhibit noise. However, this is not a major issue for our targeted applications such as image abstraction and edge detection where the smoothing operator for reducing textures also effectively reduces noise.

The camera-flash baseline determines the effective acquisition ranges (i.e., to capture distinctive shadows). If we place the camera too far away, the shadows will be too narrow to be observed due to the small baseline. On the other hand, if we place the camera too close to the object, the LED cannot cover the complete region where the camera is imaging as the LED beam has a relatively small FoV. In practice, we find that the suitable distance for acquiring an object is about 6" to 10" and the object to background distance is about 2" to 3". For example, assume the camera-object distance is 9" and the object background distance is 2.5", reusing the derivation from¹ we can obtain that the shadow width in the image is about 9 pixels on the iPhone 4S camera which uses a focal length of 0.17". Further, if the width of the object is smaller than 0.14", the shadows can appear detached.

6.2 Imaging Application

Figure 4 shows the MF results on a 6" cowboy model in front of a white background. We acquire the images with the device held by hand. Fig. 4(a) shows one of the LED flashed image and Fig. 4(b) shows the extracted depth edges. Compared with Canny edge detection (Fig. 4(c)), the MF edge map is of much better quality despite slight hand moves. The results after image registration are further improved as shown in Fig. 4(d). We observe though a spurious edge appear on the hat of the cowboy which is caused by detaching shadows due to the small size of the hat. Fig. 4(e) and (f) show various non-photorealistic rendering effects. The color of the scene is

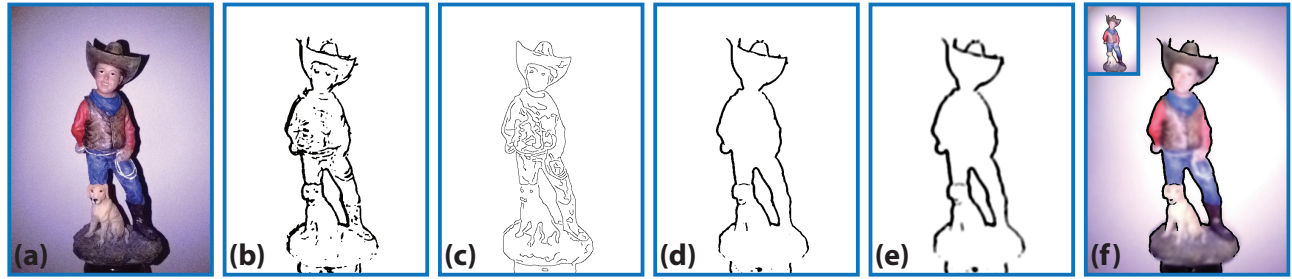


Figure 4. (a) The shadowed image. (b) Extracted depth edge image before image registration. (c) Detected depth edge image using Canny edge detector. (d) Extracted depth edge image after image registration and translation. (e) Line-art Rendering. (f) Image abstraction and image thumbnailing.

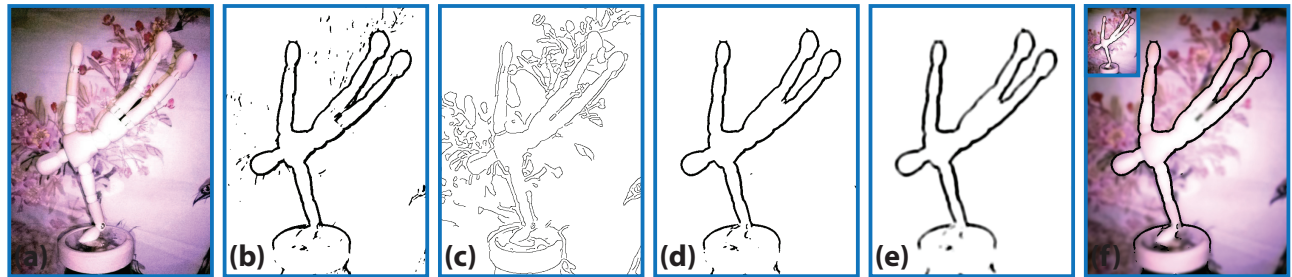


Figure 5. (a) The maximum composite image. (b) Extracted depth edge image before image registration. (c) Detected depth edge image using Canny edge detector. (d) Extracted depth edge image after image registration and translation. (e) Line-art Rendering. (f) Image abstraction and image thumbnailing.

also washed out by the flash and we normalize the maximum composite color images using linear mapping to enhance the color.

Figure 5 demonstrates using our mobile MF camera on a headstand mannequin of 5.5" in height. The mannequin is placed in front of a highly textured background to illustrate the robustness of our technique. The textures on the background provide useful features for registering images captured by our hand-held device. Fig. 5(b) and (d) show the depth edge results with and without image registration. Despite some spurious edges caused by the specular pedestal, our recovered occlusion contours are generally of good quality. Our technique fails though to capture the inner contour of the legs of the model. We observe in the maximum image that this area was not well illuminated by any of the four flashes, as shown in Fig. 5(a). The problem, however, can be alleviated by using more flashes.

In Fig. 6, we show using mobile MF for acquiring a complex plant that are covered by leaves and branches. The scene is challenging for traditional stereo matching algorithms because of heavy occlusions and high similarity between different parts of the scene. Previous SLR-camera based MF systems¹ have shown great success on recovering depth edges on such complex scenes but it uses a bulky setup (Fig. 1) and bright flashes. Our mobile MF camera produces comparable results as shown in Fig. 6(b). The thin tip of the leaves cause detached shadows and leads to splitting edges, an artifacts commonly observed in MF-based techniques.

Figure 7 demonstrates the potential of using our mobile MF to enhance human-device interactions. We use the mobile MF device for acquiring the contour of hands. Fig. 7(c) and (e) compares the foreground segmentation vs. our MF-based edge extraction. As the hand and the background shirt contain similar color and textures, segmentation based method fails to obtain accurate hand contours. In contrast, our mobile MF technique faithfully reconstructs the contours and the results can be used as input to gesture-based interfaces. One downside of our technique though is that the flashes cause visual disturbances. The problem can be potentially resolved by coupling infrared LED flashes such as 1W 850 nm infrared LED from Super Bright LEDs and the infrared camera that is already available on latest mobile devices.

6.3 Visual Inference Application

For object category classification, we created a dataset containing 5 categories similar to the *Category-5* dataset used in.¹² Each of the 5 categories contains 25 images (accompanied with depth edge images) taken from 5 objects. For each object, images are taken from 5 poses (0°, 90°, 135°, 180°, 270°) with 5 different background. Each image is generated along with depth edges using the proposed mobile multi-flash camera.

Standard bag-of-visual-word (BOW) and BOW with depth edge filtering (BOW+DE) are compared to evaluate the effectiveness of proposed camera. Training and testing sets are randomly divided into half for each run and the experimental result is summarized over 100 such random splits. The performance of BOW and BOW+DE are reported in terms of recognition rate in Table 1.

Method	BOW	BOW+DE
Classification Accuracy (%)	66.52± 4.85	75.42 ± 3.42

Table 1. Category classification result.

The result has shown that using depth edge images has significant improvement (about 10%) in recognition rate. This result is consistent with that found in.¹² It suggests that the proposed mobile multi-flash camera shares the similar performance with traditional multi-flash camera system but it is much compact and light-weighted.

7. DISCUSSIONS AND FUTURE WORK

We have presented a new mobile multi-flash camera that directly uses the mobile device's own flash as a pseudo synchronization unit. Our mobile MF camera is compact, light-weight, and inexpensive and can be mounted on most smart phones and tablets as a hand-held imaging system. To process the MF images, we have exported the OpenCV library onto mobile platforms and have developed a class of imaging processing algorithms to register misaligned images due to hand motions, extract depth edges by analyzing shadow variations, and produce non-photorealistic effects. Our solution showcases the potential of exporting computational photography techniques onto mobile platforms.

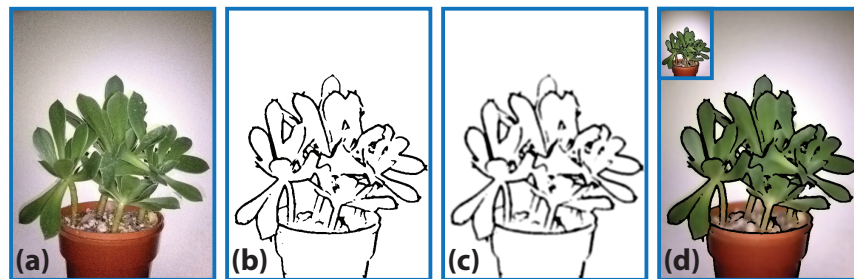


Figure 6. (a) The maximum composite image. (b) Extracted depth edge image. (c) Line-art Rendering. (d) Image abstraction and image thumbnailing.

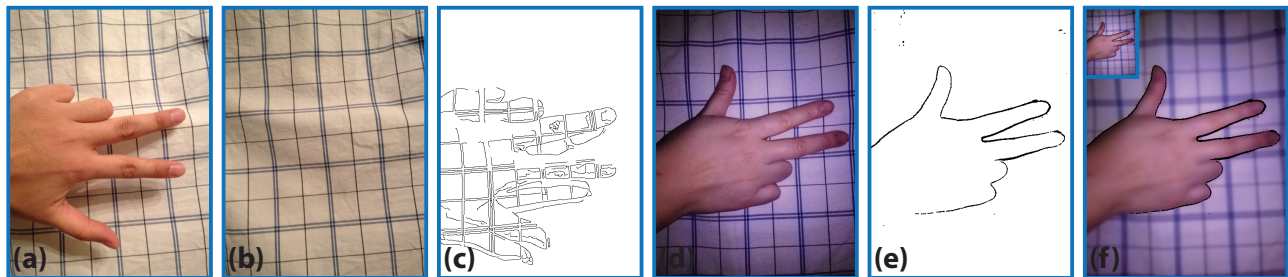


Figure 7. (a) The foreground image. (b) The background image. (c) Foreground contour from foreground-background subtraction. (d) One shadowed image. (e) The depth edge image. (f) Image abstraction and image thumbnailing.

In addition to improving the hardware designs of our system such as incorporating infrared flashes and using more flashes, we plan to explore a number of future directions. On the computer vision front, we show that the depth edge image generated on the proposed system will enhance the performance of the object category classification. Next we plan to investigate improving detection, tracking, and recognition on mobile devices. On the graphics front, we have demonstrated using mobile MF camera for photo cartooning, abstraction, and thumbnailing. Our immediate next step is to explore a broader range of image manipulation applications including depth edge guided image retargeting¹⁸ and distracting regions de-emphasis.¹⁹ Finally, on the HCI front, we will investigate using our mobile MF technique for enhancing hand gesture and head pose based schemes.

8. ACKNOWLEDGEMENT

This project is supported by the National Science Foundation under grant 1218177.

REFERENCES

- [1] Raskar, R., Tan, K.-H., Feris, R., Yu, J., and Turk, M., “Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging,” in *[ACM SIGGRAPH 2004 Papers]*, *SIGGRAPH '04*, 679–688, ACM, New York, NY, USA (2004).
- [2] Eisemann, E. and Durand, F., “Flash photography enhancement via intrinsic relighting,” in *[ACM SIGGRAPH 2004 Papers]*, *SIGGRAPH '04*, 673–678, ACM, New York, NY, USA (2004).
- [3] Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., and Toyama, K., “Digital photography with flash and no-flash image pairs,” in *[ACM SIGGRAPH 2004 Papers]*, *SIGGRAPH '04*, 664–672, ACM, New York, NY, USA (2004).
- [4] Krishnan, D. and Fergus, R., “Dark flash photography,” *ACM Trans. Graph.* **28**, 96:1–96:11 (July 2009).
- [5] Feris, R., Raskar, R., Chen, L., Tan, K.-H., and Turk, M., “Multiflash stereopsis: Depth-edge-preserving stereo with small baseline illumination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**, 147–159 (Jan. 2008).
- [6] Liu, M.-Y., Tuzel, O., Veeraraghavan, A., Chellappa, R., Agrawal, A. K., and Okuda, H., “Pose estimation in heavy clutter using a multi-flash camera,” in *[IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, Alaska, USA, 3-7 May 2010]*, 2028–2035, IEEE (2010).
- [7] Vaquero, D. A., Feris, R. S., Turk, M., and Raskar, R., “Characterizing the shadow space of camera-light pairs,” in *[IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)]*, (June 2008).
- [8] Feris, R., Turk, M., and Raskar, R., “Dealing with multi-scale depth changes and motion in depth edge detection,” in *[Computer Graphics and Image Processing, 2006. SIBGRAPI '06. 19th Brazilian Symposium on]*, 3–10 (Oct. 2006).
- [9] Taguchi, Y., “Rainbow flash camera: Depth edge extraction using complementary colors,” in *[Computer Vision ECCV 2012]*, Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C., eds., *Lecture Notes in Computer Science* **7577**, 513–527, Springer Berlin Heidelberg (2012).
- [10] Kim, Y., Yu, J., Yu, X., and Lee, S., “Line-art illustration of dynamic and specular surfaces,” *ACM Transactions on Graphics (SIGGRAPH ASIA 2008)* **27** (Dec. 2008).
- [11] Marchesotti, L., Cifarelli, C., and Csurka, G., “A framework for visual saliency detection with applications to image thumbnailing,” in *[Computer Vision, 2009 IEEE 12th International Conference on]*, 2232–2239 (29 2009-oct. 2 2009).
- [12] Sun, J., Thorpe, C., Xie, N., Yu, J., and Ling, H., “Object category classification using occluding contours,” in *[Proceedings of the 6th international conference on Advances in visual computing - Volume Part I, ISVC'10]*, 296–305, Springer-Verlag, Berlin, Heidelberg (2010).
- [13] Gelfand, N., Adams, A., Park, S. H., and Pulli, K., “Multi-exposure imaging on mobile devices,” in *[Proceedings of the international conference on Multimedia]*, *MM '10*, 823–826, ACM, New York, NY, USA (2010).
- [14] Pulli, K., Tico, M., and Xiong, Y., “Mobile panoramic imaging system,” in *[Sixth IEEE Workshop on Embedded Computer Vision]*, (2010).

- [15] Davis, A., Levoy, M., and Durand, F., “Unstructured light fields,” *Comp. Graph. Forum* **31**, 305–314 (May 2012).
- [16] Adams, A., Talvala, E.-V., Park, S. H., Jacobs, D. E., Ajdin, B., Gelfand, N., Dolson, J., Vaquero, D., Baek, J., Tico, M., Lensch, H. P. A., Matusik, W., Pulli, K., Horowitz, M., and Levoy, M., “The frankencamera: an experimental platform for computational photography,” *ACM Trans. Graph.* **29**, 29:1–29:12 (July 2010).
- [17] Black, M., Sapiro, G., Marimont, D., and Heeger, D., “Robust anisotropic diffusion,” *IEEE Transactions on Image Processing* **7**, 421–432 (mar 1998).
- [18] Vaquero, D., Turk, M., Pulli, K., Tico, M., and Gelf, N., “A survey of image retargeting techniques,” in [*Proceedings of SPIE-The International Society for Optical Engineering, Applications of Digital Image Processing XXXIII*], (2010).
- [19] Su, S. L., Durand, F., and Agrawala, M., “De-emphasis of distracting image regions using texture power maps,” in [*Texture 2005: Proceedings of the 4th IEEE International Workshop on Texture Analysis and Synthesis in conjunction with ICCV’05*], 119–124 (October 2005).