Robust Stereo with Flash and No-flash Image Pairs

Changyin Zhou
Columbia University
New York, NY
changyin@cs.columbia.edu

Alejandro Troccoli
NVIDIA Research
Santa Clara, CA
atroccoli@nvidia.com

Kari Pulli
NVIDIA Research
Santa Clara, CA
karip@nvidia.com

Abstract

We propose a new stereo technique using a pair of flash and no-flash stereo images that is both efficient and robust in handling occlusion boundaries. Our work is motivated by the observation that the brightness variations introduced by the flash can provide a robust cue for establishing stereo matches at occlusion boundaries. This photometric cue is computed per pixel, and though on its own is not robust to reliably resolve depth, it can provide a new discriminant to support patch-based stereo matching algorithms. Our experiments using a hand-held Fujifilm W3 3D camera show satisfying stereo performance over a variety of scenes, including several outdoor scenes.

1. Introduction

Equipping computers with stereo cameras and computing depths and spatial relations from stereo image pairs has been a well-researched problem [22], with applications in 3D modeling for robot navigation, new image synthesis, augmented reality, gaming and many others. Recently, stereo imaging has found its way to consumer products such as digital cameras, mobile phones, and tablets (see Figure 1). The basic stereo reconstruction algorithm works by finding the projection of the same scene point over two or more images. Under the assumption that each scene point projects the same brightness on each image, stereo reconstruction becomes a pixel matching problem. The matching is typically done by minimizing sums of squared or absolute differences, maximizing pixel correlations, or by applying a rank or census transform [27] and then matching the ranks or bit strings. This works fairly well if there is enough texture, but surfaces with uniform color are challenging [2]. In addition, pixel matching can be difficult at occlusion boundaries caused by depth discontinuities because the local region around a boundary pixel will be different. This has been addressed by using multiple shifted windows and choosing the one that matches the best, with the assumption that all those pixels come from the foreground object [4]. Even this heuristic can fail with thin objects.

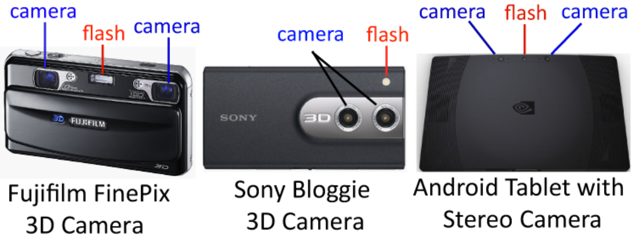


Figure 1. Consumer stereo cameras with a flash.

The basic, local algorithm, performs a winner-takes-all (WTA) approach that assigns to each pixel the disparity providing the best match. The result is obtained quickly, but is usually noisy. Global methods, in contrast, are generally formulated using an energy minimization framework to find the best match over all the image pixels imposing some smoothness conditions. These tend to be computationally expensive. More efficient approximations to the global solution have been sought using dynamic programming, computing the total cost in stages and reusing subcomputations that would otherwise be repeated [3, 4, 7]. Whereas dynamic programming methods generally fail to simultaneously incorporate both horizontal and vertical continuity constraints, graph cuts [14] succeed by formulating the stereo matching problem as finding the minimum cut in a graph, which can be thought of as extending dynamic programming to three dimensions. Another global approach is to cast the matching problem into a Markov network framework and solve it using belief propagation [24]. More details and comparisons on the state of the art algorithms are given in [22, 5, 12].

Pixel correspondences are not the only cues that can be used in 3D reconstruction. Shading provides a different cue for surface shapes and therefore relative depths, and allows reconstructing the shape of a surface from its intensity variations [6, 13, 28]. Most such methods include simplifying assumptions, such as Lambertian surface reflectance. Photometric stereo uses a minimum of three light sources that can be selectively turned on and off to recover the albedo and the surface orientation at every pixel. The light sources play a role similar to the two cameras in traditional stereo. Samaras *et al*. [21] combine shape from shading and stereo for 3D reconstruction. In binocular photometric stereo, Du *et al*. [9] integrate parallax and shading cues under a sequence of illumination directions to obtain both metric depth and fine surface details. In addition, shadows can provide a hint for relative depths [23, 15].

Active light systems used in 3D acquisition project a pattern on the scene to create a synthetic texture allowing accurate matches in areas of constant color [8, 20]. The Microsoft Kinect uses an infra-red projector-camera system to resolve the depths in real-time for objects up to a distance of 6m. Anderson *et al*. [1] project three colored lights to a scene, capture the image with the color camera in Kinect 3D system, and combine the normals from photometric stereo with Kinect's stereo depth map, producing higher quality depth maps than from Kinect alone. Our proposed flash-stereo system uses active illumination, with the advantage that it does not require to calibrate the flash position and intensity, and can even work in the outdoors.

In computational photography, flash and no-flash pairs have been used to obtain better detail, and to create shadows that can be used as cues to separate foreground from the background. Petschnigg *et al*. [18] capture flash and no-flash image pairs and propose to transfer details in flash images to the noisy no-flash images. Raskar *et al*. [19] use four flashes to detect depth discontinuities based on the projection of shadows and create stylized images. Feris *et al*. [11] propose using small baseline multiple-flash illumination to assist detecting and preserving depth discontinuity in stereo vision. Both approaches require strong illumination, short distances to the foreground objects, and short distance to the background behind them in order to create obvious shadows, limiting their applicability.

Our flash/no-flash system. Armed with these ideas we set ourselves to build a stereo system using a consumer 3D camera and two sets of stereo images: one under ambient light and one with ambient plus flash light. In the onset of our work we hoped that by carefully calibrating the flash location and intensity, we would get additional constraints for both, the object surface orientation based on the flash reflectance variations, and for the distance from the flash by estimating the fall-off of the flash intensity. We further expected to use the shadows cast by the flash and imaged by the two cameras to detect the object boundaries and the distance between a foreground object and the background over which the shadow is cast.

Our first tests on the scene in the left of Figure 2 were promising, we were able to see all the cues we expected, such as light variations and shadows. However, other more realistic scenes, like the hand scene in Figure 2 did not work as well: calibration of the flash radiance as well as completely separating its contribution from that of the ambient light proved very difficult, and since we had a single light source, we were unable to separate the effects of surface depth and orientation from the effects of varying surface reflectance. Another simple approach that one may think of is to combine each flash/no-flash RGB image pair into one 6D image and then applying a traditional stereo algorithm. This approach unfortunately does not work well mainly because adding extra 3D dimensions alone helps very little in discriminating depth discontinuities.

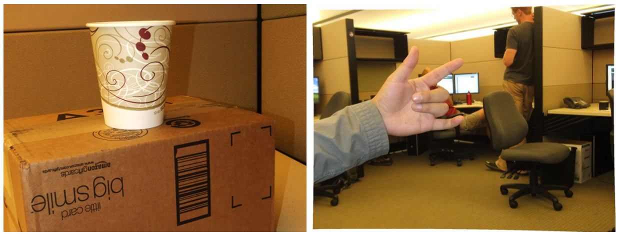


Figure 2. Our first scenes: a cup on a box and a hand.

However, we found that we could use the ratio of flash/no-flash images as an additional per-pixel constraint for stereo matching; a constraint that *is* robust against changes in illumination, surface reflectance, and does not require knowledge of the flash position or its intensity, as long as the flash is sufficiently close to the cameras. The ratio images provide a strong per-pixel cue allowing reliable matching at occlusion boundaries, and even on narrow foreground object structures, with which traditional window-based stereo matching approaches struggle. The result is a disparity map that is much cleaner than what simple winner-takes-all approaches produce.

To keep matching robust and fast, we compute the disparities at the resolution of integer pixels, which quantizes the depth estimates especially for distant objects. We post-process the basic stereo result to reduce quantization and provide a smoother output. The results are as good or better as many methods that use global optimization, but the processing requirements, both in time and memory, are much lower, making the approach practical for implementation in hand-held devices such as tablets and mobile phones.

2. Flash / No-Flash stereo

This section details our stereo matching pipeline. We first describe our mathematical framework and justify the properties of the ratio image R the we use in our matching process. After stereo matching and left-right consistency check we smooth the disparity images using the R map. We finally discuss the computational and memory complexity of our method and some alternatives.

2.1. Image formation

Figure 3 illustrates a binocular system with two identical parallel cameras located at $O_l = (-b/2, 0, 0)$ and

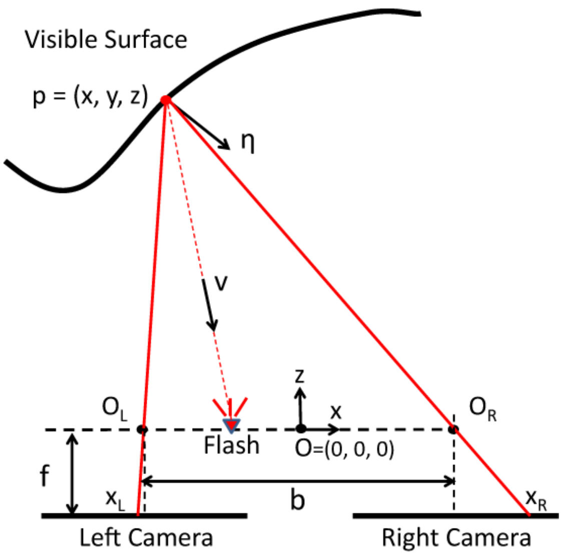


Figure 3. The geometry of a binocular stereo camera with a flash.

$O_r = (b/2, 0, 0)$, both aligned along the positive z-axis, and with lens focal length f. The flash is located somewhere close to the cameras, but its precise location need not be known. Our input data consists of two pairs of stereo images, the first pair (F_l and F_r) with flash, and a second pair (G_l and G_r) without flash. We rectify the images so that any surface point visible to both cameras projects to the same scanline. For a scene point p, we have $x_l(p) = x_r(p) - D(p_z)$, where $x_l(p)$ and $x_r(p)$ are the x-coordinates of p's projection to the left and right images and $D(p_z) = x_l(p) - x_r(p)$ is the disparity that depends on the depth at p: $D(p_z) = b \cdot f/p_z$. Therefore, for surface points p visible to both cameras, we have $[F_l, G_l](x) = [F_r, G_r](x + D(p_z))$. Stereo vision techniques estimate the disparity map D by fixing one image as reference and finding for each pixel a corresponding pixel in the other image, and storing the disparity. From the disparities we can find the depth at each pixel as $p_z = b \cdot f/D$.

For Lambertian surfaces, the measured intensity of p in the no-flash images G relates to ambient illumination, surface shape, and reflectivity by

$$G(p) = \eta \cdot I_a \cdot R_s, \tag{1}$$

where I_a is the intensity of ambient illumination (at p, omitted for clarity), R_s is the surface reflectivity (again, at p), and η is a proportionality constant between measured irradiance and scene radiance. With flash on, we have

$$F(p) = \eta \cdot I_a \cdot R_s + \eta \cdot I_f \cdot \frac{\langle \hat{n}, \hat{v} \rangle}{r^2} \cdot R_s, \tag{2}$$

where I_f is the intensity of the flash, $\langle \hat{n}, \hat{v} \rangle$ is the inner product between the surface normal and direction to the flash, and r is the distance from p to the flash.

By dividing Equation 2 by 1 and taking log, we get the ratio

$$R(p) = \log \frac{F(p)}{G(p)} = \log \left(1 + \frac{I_f}{I_a} \cdot \frac{\langle \hat{n}, \hat{v} \rangle}{r^2} \right). \tag{3}$$

We can see that this ratio image R is independent of surface reflectivity, and varies based on the surface normal and object distance. Note that this independence still holds even if the exposures of flash and no-flash image, t_l, t_r are different and even if Gamma correction has been applied:

$$R(p) = \log \frac{(t_l \cdot F(p))^\gamma}{(t_r \cdot G(p))^\gamma} = \gamma \left(\log \frac{t_l}{t_r} + \log \frac{F(p)}{G(p)} \right) \tag{4}$$

is still independent of surface reflectivity. To avoid division by zero, we define the ratio image as $R = log(F + \epsilon) - log(G + \epsilon)$, where ϵ is a small number.

2.2. Flash/no-flash for stereo matching

Equations 3 and 4 show that the ratio map R is essentially independent of the scene albedo and is instead related to distance and surface orientation. Although this equation will not be accurate for non-Lambertian surfaces, our key observation is that neighboring pixels with similar R values are likely to originate from the same surface, and neighboring pixels with very different R values are either at different distance or have different surface orientation.

We make use of this particular property of R images to solve the well-known difficulties at depth discontinuities in stereo matching. While the traditional stereo methods use a fixed patch shape for correspondence matching, we propose using the R map to find proper patch shapes for stereo matching. As shown in Figure 4 (a,b), the traditional stereo uses rectangular or circular patches for matching. At occlusion boundary, such patch contains two different depths and often leads to wrong disparity estimates. However, our technique constrains only foreground pixels to be included in the matching cost by incorporating R into the selection weight, yielding much more accurate and sharp disparity maps.

For each pixel x in the left image, the matching cost of disparity D is computed as

$$C(x, D) = \sum_{|\Delta| < r} N_{\sigma_\Delta}(\Delta) \cdot N_{\sigma_R}(d_R) \cdot |d_F|^2 \tag{5}$$

where Δ is an offset in pixels within the extent r of the matching window, r is the maximum radius of the offset, $d_R = R(x + \Delta) - R(x)$ is the difference in the ratio image, $d_F = F_l(x + \Delta) - F_r(x - D + \Delta)$ is the difference in the flash stereo pair, and $N_\sigma(\cdot)$ is a Gaussian of standard deviation σ. A small σ_Δ indicates that the extent of the neighborhood is small, and near neighbors are always weighted more than farther ones. σ_R determines how strongly the ratio image information is used to guide the matching. In our

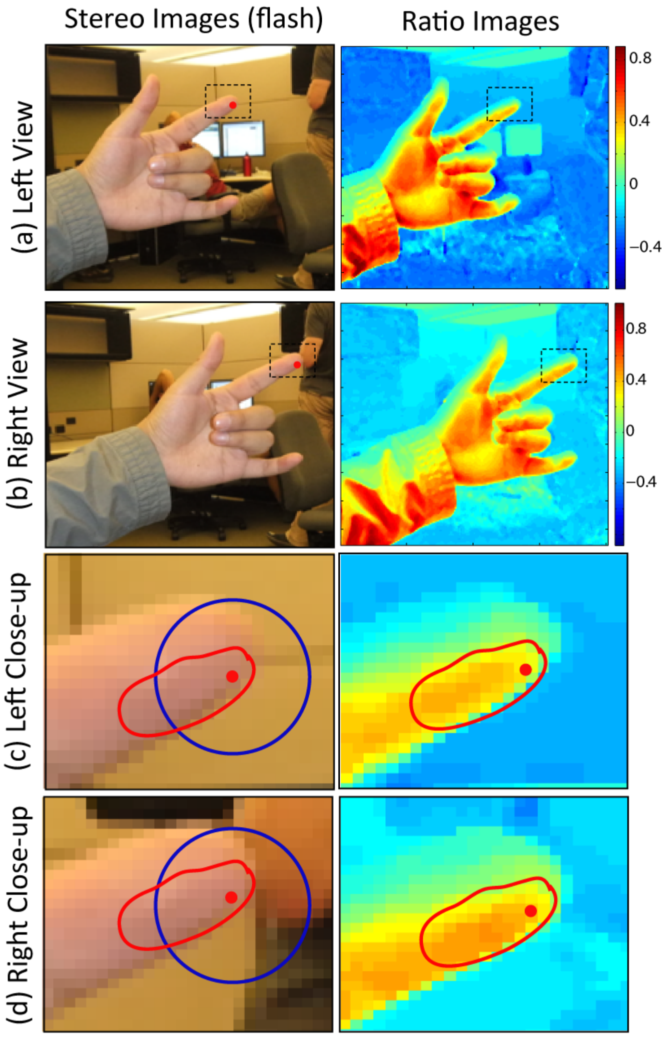


Figure 4. Traditional fixed neighborhoods around a matching point close to occlusion boundary have different backgrounds in the left (a) and right (b) images. Our cross-bilateral weight that multiplies distance term (blue) and ratio term (red) compares mostly only foreground pixels (c,d).

implementation, σ_R is chosen as a fraction of the standard deviation of R values in a local region.

The cost for the right image is computed in a similar way, only the disparity D is added to x in F_l instead of being subtracted from x in F_r. $C(x, D)$ is often referred to as a Disparity Space Image (DSI) [4].

This approach (Eqn 5) is related to the cross-bilateral filtering algorithm [18]: the more similar the neighbor's R value is to the R value at the current pixel, the more the neighbor is taken into account in determining the matching cost. Note that since we are not performing a weighted interpolation, we do not have to normalize dividing by the sum of weights, which allows faster computation. The R value of a foreground pixel close to the occlusion boundary is likely different from that of a neighboring background pixel, whose contribution is therefore mostly ignored when computing the matching cost. This behavior produces much better matches close to the occlusion edges than the traditional approaches (see Figure 6).

2.3. Winner-takes-all and Left-Right-Consistency

There are many choices for computing a disparity map from the DSI $C(x, D)$ [22]. Global optimization techniques like graph-cut and belief propagation can give better results, but they are much more expensive both in computation and memory requirements than the simple winner-takes-all (WTA) technique. Dynamic programming is relatively efficient as far as the global optimization methods go, but still increases the computation and memory requirements significantly, especially if the information is to be propagated across scanlines. We are particularly interested in trying to achieve real-time stereo and therefore choose the simplest WTA strategy with

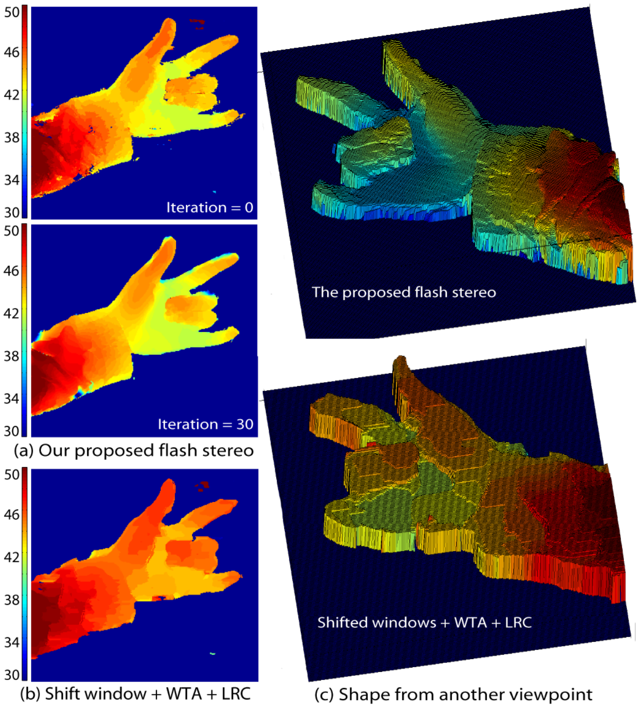
$$D(x) = \arg\min C(x, D).$$

Normally WTA produces depth results, which are either noisy or blurry, but as we show in the next section, our method aided by the ratio map produces surprisingly good results even with this simple approach. Higher quality global methods such as Graph-cut can be of course combined with our basic approach.

In order to detect surfaces that are not visible to both cameras (called occluded or semi-occluded regions), and further improve depth quality, we use the Left-Right-Consistency (LRC) technique [25, 16]. Egnal and Wildes [10] compare LRC with some other occlusion detection techniques and show that LRC consistently offers an accurate occlusion labeling. Given D_l and D_r, the two disparity maps estimated from the views of the left and right cameras, we define the consistency error as $E(x) = |D_l(x) + D_r(x+D_l(x))|$. If D_l and D_r are precisely estimated at visible points, they should be exactly opposite and yield zero consistency error. In our implementation, once $E(x) > 5$, stereo depth at pixel x will be labeled as unreliable (or occluded). For $E(x) \leq 5$, we average the disparities from the two views.

2.4. Filtering to reduce depth quantization

We compute disparities at the resolution of integer pixels, because sub-pixel disparity computation not only is less reliable, but also can significantly increase the computational cost. We propose a simple algorithm to increase the disparity precision. The basic idea is to smooth the disparity map by using similar R values as a guide. Our observation is that both the disparity and R are locally linear for any planar surface. As a result, averaging coarse but roughly correct neighboring disparities when their pixels have similar R values improves the accuracy of the disparity map.



(a) Our proposed flash stereo

(b) Shift window + WTA + LRC

(c) Shape from another viewpoint

Figure 5. Iteratively smoothing disparities along similar colors, disparities, and ratios produces better disparity maps.

However, surfaces at different depths or orientations must not be blended together.

Given the initial disparity map $D(x)$ calculated using WTA and LRC, we iterate the disparities with

$$D^{i+1}(x) = \frac{1}{\Omega} \sum_{|\Delta|<\tau} W(x, \Delta) \cdot D^{i}(x + \Delta), \qquad (6)$$

where τ is set to 2 in our implementation and Ω is the sum of all weights

$$W(x, \Delta) = N_{\sigma_R}(\delta F(x)) \cdot N_{\sigma_D}(\delta D^{i}(x)) \cdot e^{-C(x+\Delta)} \quad (7)$$

with $\delta F(x) = F(x + \Delta) - F(x)$, $\delta D^{i}(x) = D^{i}(x + \Delta) - D^{i}(x)$, and $e^{-C(x+\Delta)}$ is the confidence level of the match. The basic idea is that the depth information should flow from high confidence pixels to the low confidence ones, and that the more similar the ratio and stereo images, the stronger the flow.

We iterate this process; our experiments show that 10-30 iterations are enough for good results, and beyond that the disparity changes very slowly. In Figure 5, we zoom in to the disparity range from 40 to 51 pixels. We can see that at iteration 0, where no filtering has been applied, the disparity map of the hand is not as smooth as we would like, but smoothness improves with the number of iterations.

2.5. Computational complexity

The conventional similarity matrix computation has a computational complexity of $O(k\,N\,M)$, where k is a small factor related to the size of the accumulation kernel, N is the number of pixels in the image, and M is the maximum disparity, and requires a memory space of $O(N\,M)$. Winner-takes-all strategy does not need to store the whole DSI, and therefore reduces the memory requirement to $O(N)$.

Global optimization techniques like graph-cut require the whole DSI to be pre-computed and stored, and the computational complexity can be as high as $O(N\,\log(N)\,M)$. Techniques like graph-cut involve Singular Vector Decomposition (SVD) over a large $N \times N$ matrix, and therefore further require lot of memory, even if sparse matrix techniques are used. Yang *et al.* [26] study a fast GPU implementation of belief propagation and demonstrate a real-time stereo system, but only at very low image and disparity resolutions. Dynamic programming (DP) is known to be efficient, $O(s\,M\,N)$, where s is proportional to the number of possible state transitions. DP requires $O(s\,M\,N)$ space for the DSI for tracking the node states in optimization. The main limitation of DP is that it can efficiently only enforce depth coherence along the scanlines, not across, and it often produces obvious depth stripe artifacts.

Our proposed stereo technique (Equation 5) is essentially a cross-bilateral filter. Several fast bilateral filtering techniques have been proposed in recent years. Paris and Durand [17] show a 3D Kernel technique for fast bilateral filtering which can be done in $O(N + \frac{N\,L}{\sigma_N^2\,\sigma_L})$, where L is the number of image gray levels, and σ_N, σ_L are the 3D sampling rates. They show that the filtering quality can still be good with large σ_N, σ_L, and that by using an on-the-fly downsampling technique, the memory requirement can be kept as low as $O(N)$.

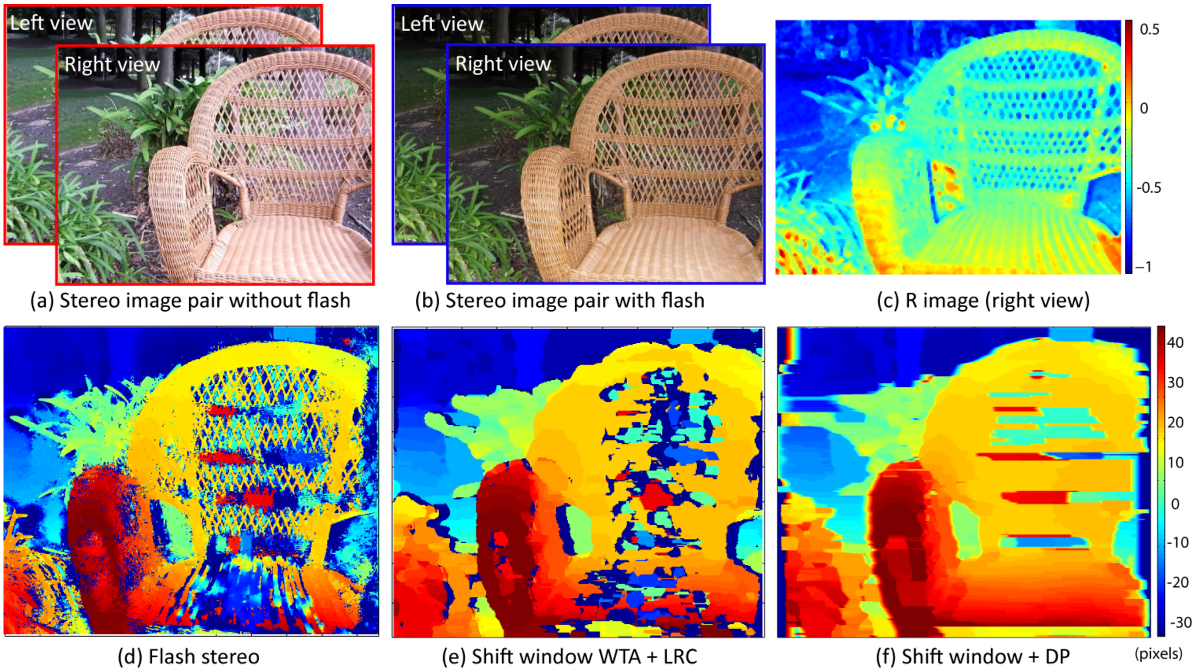
3. Experimental evaluation

We evaluate our method with various scenes, both indoors and outdoors, and compare our method against standard stereo processing algorithms we implemented.

3.1. Benchmarking our results

Our method is quite unique in the requirement of a flash and no-flash pair of stereo images as inputs, and therefore cannot be directly evaluated using publicly available images such as the Middlebury stereo dataset [22]. Instead, we captured a variety of scenes using a Fujifilm FinePix Real 3D W3 camera. This camera has two 10 MP sensors with a baseline of 75mm and is designed to capture stereo images to be displayed on a 3D television or its built-in autostereoscopic display. We did the standard stereo rig camera calibration using using Bouguet's Calibration Toolbox.[1] All of captured images are rectified before stereo matching using the precomputed and stored calibration data so that we can simply traverse scanlines during the matching.

[1] www.vision.caltech.edu/bouguetj/calib_doc/



(a) Stereo image pair without flash (b) Stereo image pair with flash (c) R image (right view)

(d) Flash stereo (e) Shift window WTA + LRC (f) Shift window + DP

Figure 6. A wicker chair captured from a distance of $2m$ on a cloudy day. The back of the chair shows a cross-shaped repetitive pattern with thin stripes. This kind of pattern represents a challenge to stereo algorithms due to repetitiveness and occlusions.

The Fujifilm 3D camera provides a convenient flash/no-flash shooting mode, where the camera captures two stereo pairs in succession with a shutter lag of a few tenths of a second. All the images for our experiments were captured by leaving most settings (including focus, exposure, f-Number, ISO, and white balance settings) in automatic mode.

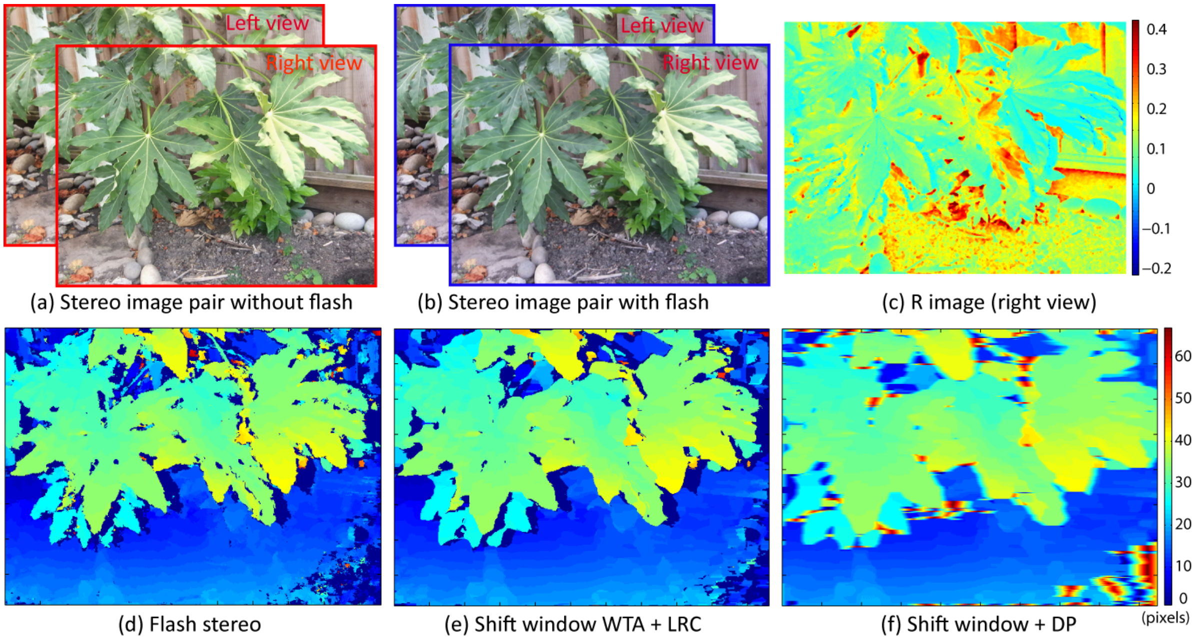
We compare our flash stereo algorithm against two traditional techniques. For all three methods we use the sum-of-squared-differences (SSD) cost metric. Both of the comparison methods use the shift-window approach to improve their behavior close to occlusion edges. Both our method and one of the comparison methods use WTA and LRC strategies, while the other one uses 4-state dynamic programming (DP) [7]. Our method used $\sigma_R = 0.01$ and $\sigma_D = 3$. For each scene, our technique makes use of all four images; the benchmark techniques only use the no-flash stereo images.

3.2. Stereo matching results

Wicker chair. We first evaluate our algorithm on a scene with a wicker chair shown in Figure 6, captured from a distance of $2m$ on a cloudy day. The back of the chair shows a cross-shaped repetitive pattern with thin stripes. This kind of pattern represents a challenge to stereo algorithms due to repetitiveness and occlusions. Figure 6 (d) shows the depth map obtained using our proposed flash stereo methodology, 6 (e) shows the depth map computed using sums-of-squared-distances with shifted windows WTA plus LRC, and 6 (f) using shifted windows and dynamic programming in place of WTA. Our algorithm is able to resolve the occlusion boundaries very well on the back of the chair, but it fails to resolve some of the repetitive patterns on the seat. The local WTA with shifted windows shows errors both on the back and on the seat. DP failed on almost all clutter regions due to its strong assumption of smoothness, but it worked well on the seat of the chair where there are no depth discontinuities. For this scene we choose a large patch size $r = 17$ for flash stereo in order to combat the repetitive patterns. For shifted windows WTA and we also tried different patch sizes and got the best results with $r = 11$ and $r = 10$, respectively.

Garden leaves. The second example in Figure 7 shows plant leaves in the garden, where the flash is relatively weak compared to the ambient light from the open sky. The nearest leaf is about 1.5 meter away from the camera. We can see that the ratio image varies with distance and surface orientation. Compared against the result of the other two benchmark techniques, our proposed flash stereo technique (d) reveals better 3D structures of the leaves. Notice that thin stems are also well reconstructed. Dynamic programming (e) poses strong constrains on the surface smoothness and therefore suffers from severe blurriness in the disparity map.

Cup on a box. Our final example in Figure 8 shows a simple indoor scene with a cup on a box on an office desk. The object surface has little texture, which makes the problem difficult. We can see that dynamic programming can recover the disparity of the non-textured region better by using global optimization, but yields blurry results along the depth edges. As expected, the two WTA methods can-



(a) Stereo image pair without flash (b) Stereo image pair with flash (c) R image (right view)

(d) Flash stereo (e) Shift window WTA + LRC (f) Shift window + DP

Figure 7. A garden scene captured under morning sunlight. The nearest leaf is about 1.5 meter away from the camera.

not recover the depth of non-textured surface well, but they perform better at the sharp depth edges. Of these two, our technique seems to work better on this scene.

4. Conclusion

We have presented an efficient and robust stereo system for use with consumer stereo cameras. Our system uses a flash that is available on all such stereo systems, and it works without requiring calibration of the flash position or radiometric quantities such as its intensity or frequency distribution. It even works outdoors, as shown in one of the example scenes. This is in contrast to several shape and range cues we originally hoped to work, such as distance-based light fall-off (which would have required flash calibration), modulation of the reflectance based on surface normals and using those as actual normal constraints, or using shadows as cues for distance between the occluding and occluded surfaces.

The ratio image of the flash and no-flash exposures provides a robust per-pixel constraint that helps in determining correct pixel correspondences. In particular, when combined with cross-bilateral filtering, it avoids mixing in background pixels when matching foreground object pixels, yielding crisp occlusion boundaries. Compared to shifted windows, it can also be applied for matching foreground objects thinner than the width of a matching window.

There are limitations with the proposed approach. First, it benefits from ratio image constraints only when the foreground objects are within the reach of the flash and the ratio image shows discontinuities at depth boundaries. The proposed technique strongly prefers flashes with more power and smaller beam angle because a stronger flash light produces ratio images of higher dynamic range (or higher precision). A quantitative evaluation of the effect of flash power is worthy of further exploration. Secondly, strongly specular surfaces are always challenging for any stereo method, but since we project additional light to the scene, ratio images of specular surfaces are even more unreliable. Finally, the current method is only suitable for static scenes.

References

[1] R. Anderson, B. Stenger, and R. Cipolla. Augmenting depth camera output using photometric stereo. In *IEEE International Conference on Computer Vision*, 2011. 2

[2] S. Baker, T. Sim, and T. Kanade. When is the Shape of a Scene Unique Given its Light-Field: A Fundamental Theorem of 3D Vision? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1), 2003. 1

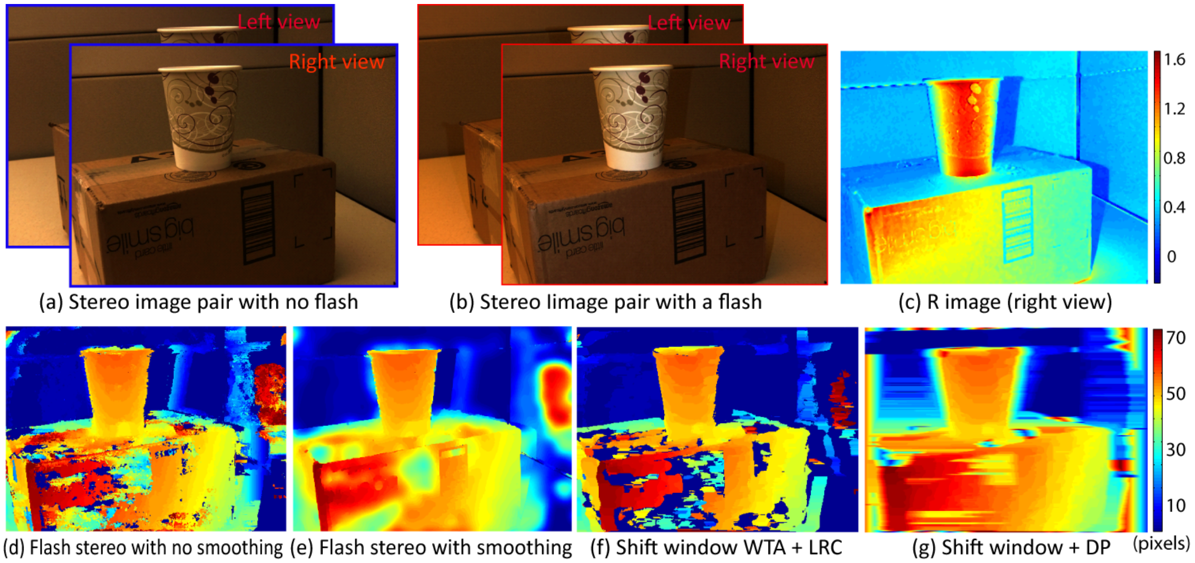
[3] P. Belhumeur and D. Mumford. A bayesian treatment of the stereo correspondence problem using half-occluded regions. In *IEEE Computer Vision and Pattern Recognition*, 1992. 1

[4] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3), 1999. 1, 4

[5] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8), 2003. 1

[6] H. Bülthoff and H. Mallot. Integration of depth modules: stereo and shading. *Journal of Optical Society of America*, 5(10), 1988. 1

[7] A. Criminisi, A. Blake, C. Rother, J. Shotton, and P. Torr. Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming. *International Journal of Computer Vision*, 71(1), 2007. 1, 6



(a) Stereo image pair with no flash (b) Stereo Iimage pair with a flash (c) R image (right view)

(d) Flash stereo with no smoothing (e) Flash stereo with smoothing (f) Shift window WTA + LRC (g) Shift window + DP

Figure 8. A cup and a box on an office table captured in a close distance.

[8] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, jun 1995. 2

[9] H. Du, D. B. Goldman, and S. M. Seitz. Binocular photometric stereo. In *British Machine Vision Conference*, 2011. 2

[10] G. Egnal and R. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8), 2002. 4

[11] R. Feris, R. Raskar, L. Chen, K. Tan, and M. Turk. Discontinuity preserving stereo with small baseline multi-flash illumination. In *IEEE International Conference on Computer Vision*, 2005. 2

[12] M. Gong, R. Yang, L. Wang, and M. Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *International Journal of Computer Vision*, 75(2), 2007. 1

[13] B. Horn. Obtaining shape from shading information. In *Shape from shading*, pages 123–171. MIT Press, 1989. 1

[14] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, 2001. 1

[15] D. Kriegman and P. Belhumeur. What shadows reveal about object structure. *Journal of Optical Society of America*, 18(8), 2001. 2

[16] A. Luo and H. Burkhardt. An intensity-based cooperative bidirectional stereo matching with simultaneous detection of discontinuities and occlusions. *International Journal of Computer Vision*, 15(3), 1995. 4

[17] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *European Conference on Computer Vision*, 2006. 5

[18] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 23(3), 2004. 2, 4

[19] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics*, 23(3), 2004. 2

[20] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Trans. Graph.*, 21, July 2002. 2

[21] D. Samaras, D. Metaxas, P. Fua, and Y. Leclerc. Variable albedo surface reconstruction from stereo and shape from shading. In *IEEE Computer Vision and Pattern Recognition*, 2000. 2

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1), 2002. 1, 4, 5

[23] S. Shafer and T. Kanade. Using shadows in finding surface orientations. *Computer Vision, Graphics, and Image Processing*, 22(1), 1983. 2

[24] J. Sun, Y. Li, S. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *IEEE Computer Vision and Pattern Recognition*, 2005. 1

[25] J. Weng, N. Ahuja, and T. Huang. Two-view matching. In *IEEE International Conference on Computer Vision*, 1988. 4

[26] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nistér. Real-time global stereo matching using hierarchical belief propagation. In *British Machine Vision Conference*, 2006. 5

[27] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision ECCV '94*, volume 801 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1994. 1

[28] R. Zhang, P. Tsai, J. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8), 1999. 1