

Disparity Estimation for Image Fusion in a Multi-aperture Camera

Janne Mustaniemi, Juho Kannala, and Janne Heikkilä

Center for Machine Vision Research
University of Oulu, Finland
{janne.mustaniemi, jkannala, jth}@ee.oulu.fi

Abstract. In this paper, an image fusion algorithm is proposed for a multi-aperture camera. Such camera is a worthy alternative to traditional Bayer filter camera in terms of image quality, camera size and camera features. The camera consists of several camera units, each having dedicated optics and color filter. The main challenge of a multi-aperture camera arises from the fact that each camera unit has a slightly different viewpoint. Our image fusion algorithm corrects the parallax error between the sub-images using a disparity map, which is estimated from the multi-spectral images. We improve the disparity estimation by combining matching costs over multiple views with help of trifocal tensors. Images are matched using two alternative matching costs, mutual information and Census transform. We also compare two different disparity estimation methods, graph cuts and semi-global matching. The results show that the overall quality of the fused images is near the reference images.

Keywords: mutual information, census transform, trifocal tensor

1 Introduction

Multi-aperture camera refers to an imaging device that comprises more than one camera unit. The camera produces several sub-images, which are combined into a single image. The main challenge of the multi-aperture camera arises from the fact that each camera unit has a slightly different viewpoint. This results to misalignment of images that needs to be corrected before images can be properly combined. In practice, the problem is solved by finding the corresponding pixels from each image.

Multi-aperture cameras can improve the image quality, camera size and camera features over the traditional single-aperture cameras. There already exist patents of such systems [1, 2]. Some of the largest mobile phone companies have also patented their versions of the multi-aperture cameras [3–5]. Probably the most complete implementations of multi-aperture camera modules come from LinX Imaging [6] and Pelican Imaging [7].

LinX Imaging has successfully developed small-sized multi-aperture cameras for mobile devices. Camera modules have two, three or four cameras and they

come in various configurations and sizes. Modules use different combination of color and monochrome cameras. Based on technology presentation in [6], captured images have higher dynamic range, lower noise levels and better color accuracy over the traditional mobile phone cameras. The height of the camera module is nearly half of a typical mobile phone camera module.

PiCam (Pelican Imaging Camera-Array) is another example of working multi-aperture camera. PiCam module consists of 4×4 array of cameras, each having dedicated optics and color filter. The final image is constructed from the low-resolution images using superresolution techniques. The image quality is comparable to existing smartphone cameras and the thickness of the camera module is less than 3 mm. [7]

An example of multi-aperture camera is shown in Figure 1. In this case, three of the lenses are equipped with red, green and blue filters. The fourth camera captures the luminance information of the scene. It may be used to increase the light sensitivity of the camera and to increase the robustness of disparity estimation. The final image is formed by combining the sub-images into a single RGB image.

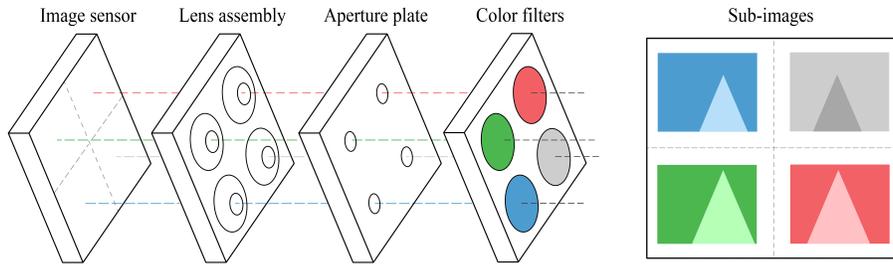


Fig. 1. Image sensing arrangement of the four-aperture camera

The thickness of the camera is closely related to the image quality the camera produces. Cameras equipped with larger image sensors typically produce better images. However, the increase in sensor size will also increase the height of the optics. Multi-aperture camera solves this problem by using a combination of smaller sensors, each having dedicated optics with reduced optical height. [3]

In Bayer filter cameras, the adjacent pixels capture the light intensity of different color bands. Consequently, the neighboring pixels may interact with each other. This phenomenon is known as crosstalk and it typically causes desaturation of color. The camera in Figure 1 does not suffer from crosstalk since each sensor is only measuring a single spectral color. [7]

Chromatic aberration is a type of distortion in which a lens fails to focus different colors to the same point on the image sensor. This occurs because lens material refracts different wavelengths of light at different angles. The effect can be seen as colored and blurred edges especially along boundaries that separate dark and bright parts of the image. The lenses in the multi-aperture camera can be much simpler since chromatic aberration does not complicate the optics

design. Besides the improved image quality, a simpler design usually means lower manufacturing costs. [7]

One of the disadvantages of the current camera phones is that they cannot produce images with shallow depth of field. Mobile phone applications such as Google Lens Blur [8] aim to address this weakness. Lens Blur captures the scene depth from the camera movement and then uses the information for post-capture refocusing. Multi-aperture camera can acquire depth information via stereo matching. Depth information is also useful in various other applications such as background removal and replacement, resizing of objects, depth based color effects and 3D scanning of objects. [6, 7]

In this paper, we propose an image fusion algorithm for a four aperture camera in Figure 1. In contrast to PiCam, we cannot match images that are captured with similar color filters. This complicates the disparity estimation since corresponding pixels may have completely different intensities in each image. Therefore, we use a robust matching cost such as mutual information or Census transform. We improve the robustness of disparity estimation over traditional two-view stereo methods such as [9, 10] by combining matching costs over four-views. We further improve the estimation by adding a luminance constraint to the cost function.

2 Image Fusion Algorithm

In this Section, an image fusion algorithm is proposed for a four-aperture camera. The processing steps of the algorithm are shown in Figure 2. Algorithm is based on disparity estimation, in which the aim is to find corresponding pixels from each image. Disparities are estimated from the multi-spectral images captured by the four-aperture camera. Parallax error between the images is then corrected using the disparity map.

2.1 Offline Calibration

For this implementation, it was chosen that I_1 is the reference image and it corresponds the image captured with green color filter. Images I_2 and I_3 correspond

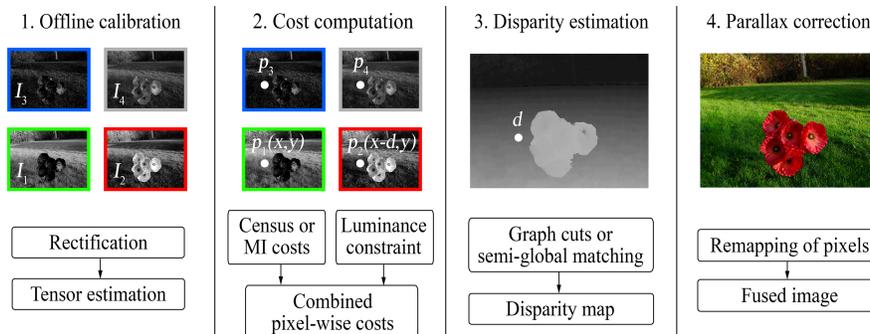


Fig. 2. Processing steps of the image fusion algorithm

to red and blue filtered images, respectively. The fourth image is used as a luminance image. The algorithm assumes that the camera movement between the first and second view is purely horizontal. This is difficult to ensure in practise, which is why image pair I_1 and I_2 is rectified. Other images are not rectified because algorithm utilizes trifocal tensors.

Trifocal tensor Image fusion can be performed by matching each image pair independently. However, such approach would not utilize the full potential of multiple views. Robustness of matching increases when matching costs from different views are combined. This will lead to a more accurate disparity map as will be demonstrated in Section 3. Consequently, the fused image will have better quality as well.

In the case of two views, a fundamental matrix is often defined to relate the geometry of a stereo pair. For three views, this role is played by the trifocal tensor. Trifocal tensor encapsulates all the geometric relations among three views. It only depends on the motion between the views and internal parameters of the cameras. Trifocal tensor is expressed by a set of three 3×3 matrices defined uniquely by the camera matrices of the views. Tensor can be constructed from the camera matrices or from the point correspondences. We used the latter approach because the camera system was uncalibrated. [11]

In practice, one can use the tensor to transfer point from a correspondence in two views to the corresponding point in a third view. This is known as point transfer. We define two trifocal tensors for each test scene. First tensor T_1 is computed for the images I_1 , I_2 and I_3 . Similarly, a second tensor T_2 is defined for the images I_1 , I_2 , and I_4 . Let assume that there is a point $p_1 = (x, y)$ in the first image and its disparity d in relation to second image is known $p_2 = (x-d, y)$. Then, the corresponding points in third and fourth image can be computed using the tensors T_1 and T_2 respectively.

2.2 Matching Cost Computation

In order to find the corresponding pixels from each image, one needs a way to measure the similarity of image locations. It is common to presume that corresponding pixels have similar intensities in all views. This assumption is often violated, in the presence of radiometric differences such as noise, specularities and reflections. Similar problems arise when cameras are equipped with different color filters. This work utilizes mutual information and Census transform similarity measures. They both are known to be robust against radiometric differences [12, 13].

To further improve the robustness of disparity estimation we use a luminance cost C_L , which is combined with mutual information or Census transform costs. Matching cost is computed at each pixel for all candidate disparities in a given disparity range. Disparity value that minimizes the cost represents the best match. The cost of assigning disparity d for pixel p is defined as follows:

$$C(p, d) = C_{MI/census} + K \cdot C_L, \quad (1)$$

where K is a constant, which controls the influence of the luminance cost C_L .

Mutual Information (MI) has been used as a similarity measure with local [13] and global [9, 10] stereo matching methods. The main advantage of MI is its ability to handle complex radiometric relationships between images. For example, MI handles matching image I_1 with the negative of image I_2 as easily as simply matching I_1 and I_2 . Mutual information of images I_1 and I_2 is defined using entropies:

$$MI_{I_1, I_2} = H_{I_1} + H_{I_2} - H_{I_1, I_2}, \quad (2)$$

where H_{I_1} and H_{I_2} are the entropies of individual images and H_{I_1, I_2} is their joint entropy. The idea of using mutual information for stereo matching comes from the observation that joint entropy is low when images are well-aligned. It can be seen from previous equation that mutual information increases when joint entropy is low.

In order to calculate the entropies, one needs to estimate the marginal and joint probability distributions of underlying images. This can be done by using a simple histogram of corresponding image parts. Joint distribution is formed by binning the corresponding intensity pairs into a two-dimensional array. The marginal distributions are then obtained from the joint distribution by summing the corresponding rows and columns.

It is possible to apply mutual information to fixed-sized windows [13]. Window-based approach suffers from the common limitations of fixed-sized windows, such as poor performance at discontinuities and in textureless regions. To overcome the difficulties of window-based approach, Kim [9] used mutual information as a pixel-wise matching cost. The computation of joint entropy H_{I_1, I_2} was transformed into a cost matrix $h_{I_1, I_2}(i_1, i_2)$, which contains costs for each combination of pixel intensities $I_1(p) = i_1$ and $I_2(p) = i_2$. In the case of two views, the cost matrix is calculated with formula:

$$h_{I_1, I_2}(i_1, i_2) = -\frac{1}{n} \log((P_{I_1, I_2}(i_1, i_2) * g(i_1, i_2)) * g(i_1, i_2)), \quad (3)$$

where $g(i_1, i_2)$ is Gaussian kernel, which is convolved with the joint distribution $P_{I_1, I_2}(i_1, i_2)$. Number of all combinations of intensities is n . Details of the derivation can be found in [9].

Cost computation is illustrated in Figure 3. The cost matrix is calculated iteratively using the disparity map from the previous iteration. At each iteration, a new disparity map is estimated based on the current cost matrix. Usually only a few number of iterations (e.g. 3 iterations) are needed until the disparity map no longer improves. First, pixels in the image I_2 are remapped based on the current disparity map. The joint distribution P_{I_1, I_2} of corresponding intensities is then calculated between the image I_1 and remapped version of the image I_2 . First iteration can use a random disparity map since even wrong disparities allow a good estimation of the joint distribution due to high number of pixels.

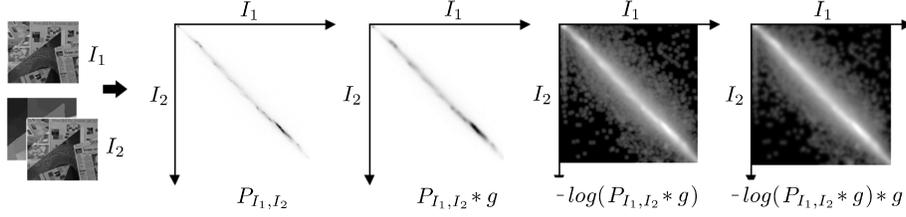


Fig. 3. Computation of mutual information cost matrix h_{I_1, I_2}

In our case, there are four images. We perform similar computations for other images, resulting to three different cost matrices h_{I_1, I_2} , h_{I_1, I_3} and h_{I_1, I_4} . Trifocal tensors are needed in order to remap images I_3 and I_4 . The matching cost of assigning disparity d for pixel p is defined as follows:

$$C_{MI}(p, d) = h_{I_1, I_2}(i_1, i_2) + h_{I_1, I_3}(i_1, i_3) + h_{I_1, I_4}(i_1, i_4). \quad (4)$$

where i_1 is the intensity of the pixel p in the first image. Intensities i_2 , i_3 and i_4 in other images depend on the disparity d .

Census transform is based on the relative ordering of local intensity values. It can tolerate all radiometric distortions that preserve this ordering [14]. Census transform maps the local neighborhood of pixel into a bit string. Pixel's intensity is compared against the neighboring pixels and the bit is set if the neighboring pixel has lower intensity than the pixel of interest. Census transform for a pixel p can be defined as follows:

$$R_p = \bigotimes_{[x, y] \in D} \xi(p, p + [x, y]), \quad (5)$$

where symbol \otimes denotes concatenation and D is the window around pixel p . The comparison operation $\xi(p, p + [x, y])$ is 1 if the neighboring pixel has lower intensity than the pixel p and otherwise 0. In this work, we use a window of 9 x 7 pixels. Each pixel in the window is compared to the center pixel. This will result to a bit string that consists of 62 bits. Computation is repeated for each of the four images.

The actual pixel-wise matching cost depends on the Hamming distance between the corresponding bit strings. Hamming distance is defined by counting the number of bits that differ in the two bit strings. For instance, the Hamming distance between two identical bit strings is zero since all bits are the same. Disparity value that minimizes the distance represents the best match. Let $H(R_{p,1}, R_{p,2})$ denote the Hamming distance between the corresponding bit strings in images I_1 and I_2 . Since there are four images in this implementation, the pixel-wise cost is a sum of Hamming distances:

$$C_{census}(p, d) = H(R_{p,1}, R_{p,2}) + H(R_{p,1}, R_{p,3}) + H(R_{p,1}, R_{p,4}). \quad (6)$$

Luminance Constraint There is an additional constraint related to the fourth image, which can be combined with mutual information or Census transform costs. Let us assume that there are four corresponding points p_1 , p_2 , p_3 and p_4 in each image. Because the fourth image represents the luminance, the corresponding points should satisfy the following equation:

$$\hat{I}_4(p_4) = G \cdot I_1(p_1) + R \cdot I_2(p_2) + B \cdot I_3(p_3), \quad (7)$$

where point's intensity is denoted by $I(p)$. Coefficients G , R and B in the previous equation depend on the color filters of the cameras. In case there is a large difference between the left and right side of the equation, it is likely that points are not correspondences. Based on this assumption, the luminance cost can be written as:

$$C_L = |I_4(p_4) - \hat{I}_4(p_4)|. \quad (8)$$

2.3 Disparity Estimation

We evaluate two different disparity estimation methods, graph cuts and semi-global matching. These methods aim to find correct disparities for every pixel in the image by using matching costs and smoothness assumptions. The idea is to favor disparity configurations in which disparity varies smoothly among neighbouring pixels.

Graph cuts method performs a global optimization process over the whole image. We employ the multi-label optimization library developed by Veksler et al. [15]. Global energy is minimized with an expansion move algorithm using the truncated absolute difference as a smoothness cost. Truncated absolute difference gave the best overall performance over Potts model.

Semi-global Matching (SGM) approximates the global energy by pathwise optimization from all directions through the image. It approximates 2D smoothness constraint by combining many 1D constraints. This work implements the semi-global block matching algorithm that is part of the OpenCV library. It is a variation of the original SGM algorithm in [10]. In contrast to graph cuts, the SGM performs post-processing steps such as subpixel interpolation, left-right consistency check and speckle filtering.

2.4 Parallax Correction

After the disparity estimation, the parallax error between the images can be corrected. In practise, pixels in the red filtered image I_2 and blue filtered image I_3 are remapped using the calculated disparity map. The green filtered image I_1 is used as a reference so there is no need to remap the image. Whereas image I_2 can be directly remapped using the disparity map, trifocal tensor is needed to remap image I_3 . After remapping, the corresponding pixels will have the same image coordinates. In case the point does not correspond to any particular

pixel, the pixels intensity is computed from neighboring pixels using bilinear interpolation.

An RGB image is then constructed by simply combining images I_1 , I_2 and I_3 . In this implementation, the luminance image I_4 is not used when forming the final image. Pixels that are located near the borders of the image may not be visible in all the images. These areas are removed from the final image based on maximum disparity parameter.

3 Experiments

The performance of the image fusion algorithm was evaluated using a test camera system. The evaluation aims to find the best combination of similarity measures and disparity estimation methods for the image fusion. Input images were captured with a traditional Bayer matrix camera, which was moved between the shots. In order to simulate the presence of different color filters, the original 24-bit RGB images were split to separate color channels. Luminance image was created from the original RGB image by weighting each color component by different amounts.

Test scenes are shown in Figure 4. Tea, Flowers and Grass datasets were captured using the same camera arrangement as illustrated in Figure 1. The baseline was approximately 12 mm for each pair of horizontal and vertical camera positions. We also used the standard Middlebury stereo datasets Teddy, Cones and Venus in which cameras are parallel to each other [16, 17]. Ground truth disparity maps were available for the images 2 and 6 in each dataset. In order to perform comparison to ground truth, we used images 2 and 6 as a first and second input image. Improved fused image could have be obtained if adjacent images were used. Image sizes and disparity ranges are listed in Table 1.



Fig. 4. Reference views for the Teddy, Cones, Venus, Tea, Flowers and Grass datasets

Fused images were compared against the original RGB images captured by the camera system. We also measured the similarity of the images using the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). SSIM values are computed for each channel of the image. Value of 1 represents the perfect match. The accuracy of the disparity estimation was evaluated by counting the number of invalid disparities in the disparity map. Disparities were not evaluated in occluded areas since occlusion handling was not implemented. Disparity was classified as invalid if its value differs more than 1 pixel from the ground truth. Smoothness parameters of the semi-global matching and graph cuts methods were manually tuned for the mutual information and Census transform costs.

Table 1. Image sizes and disparity ranges in pixels

	Tea	Flowers	Grass	Teddy	Cones	Venus
Image size	1000x745	1150x860	1024x783	450x375	450x375	434x383
Disparity range	64	32	32	64	64	32

Parameters were kept constant for Tea, Flowers and Grass datasets. Different, although constant parameters were used for Middlebury datasets.

Table 2 shows the statistics for both similarity measures when graph cuts method is used. Census transform outperforms the mutual information in all test cases if error percentages are considered. There are no significant differences in PSNR and SSIM scores.

Table 2. Results of graph cuts method

	Mutual Information			Census		
	Errors	PSNR	SSIM (rgb)	Errors	PSNR	SSIM (rgb)
Teddy	11.01	37.97	0.86; 1.00; 0.81	7.60	37.57	0.87; 1.00; 0.81
Cones	7.11	33.97	0.83; 1.00; 0.79	4.92	34.42	0.85; 1.00; 0.79
Venus	2.80	39.56	0.89; 1.00; 0.83	1.49	39.26	0.89; 1.00; 0.83
Tea	-	39.47	0.95; 1.00; 0.88	-	39.58	0.95; 1.00; 0.88
Flowers	-	39.44	0.94; 1.00; 0.86	-	39.36	0.94; 1.00; 0.86
Grass	-	33.97	0.82; 1.00; 0.84	-	34.12	0.83; 1.00; 0.85

The results of semi-global matching are shown in Table 3. As with graph cuts, the Census transform performs better than the mutual information. SGM further improves the accuracy of disparity estimation over graph cuts. PSNR and SSIM scores are also better. The main improvements come from the sub-pixel accurate disparity estimation and left-right consistency check. The resulting disparity map and fused image for the Teddy dataset is shown in Figure 5.

Table 3. Results of semi-global matching

	Mutual Information			Census		
	Errors	PSNR	SSIM (rgb)	Errors	PSNR	SSIM (rgb)
Teddy	10.92	38.43	0.88; 1.00; 0.81	6.81	38.32	0.89; 1.00; 0.81
Cones	6.84	34.95	0.86; 1.00; 0.79	4.67	35.10	0.87; 1.00; 0.79
Venus	2.96	41.22	0.91; 1.00; 0.83	1.30	40.40	0.90; 1.00; 0.83
Tea	-	40.45	0.96; 1.00; 0.89	-	40.36	0.96; 1.00; 0.89
Flowers	-	40.05	0.94; 1.00; 0.87	-	40.12	0.95; 1.00; 0.87
Grass	-	34.19	0.82; 1.00; 0.84	-	35.01	0.86; 1.00; 0.87

The advantages of using trifocal tensor and four different views are best demonstrated with disparity maps. The left most disparity map in Figure 6 is generated using only one pair of stereo images, graph cuts and Census transform. In this example, the green filtered image is matched to red filtered image. The second image is matched using green, red and blue filtered images and trifocal tensor. The third image uses all four input images but does not take advantage of the luminance constraint. Adding the luminance constraint to the cost function will further improve the disparity map as shown in the last image. Consequently, the

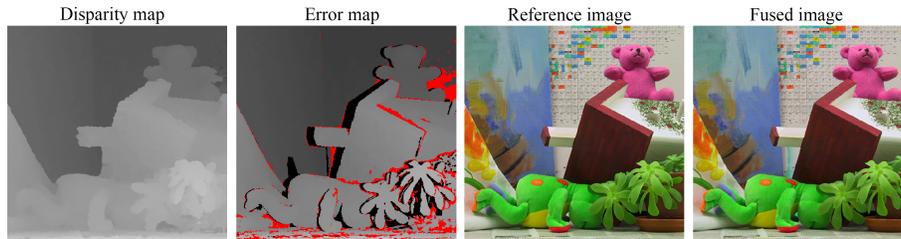


Fig. 5. The result of semi-global matching and Census transform on Teddy dataset. Red areas in the error map represent erroneous disparities and black areas are occlusions.

disparity map will also produce the best fused image. Smoothness parameter was tuned for each test so that the disparity map would be as accurate as possible.

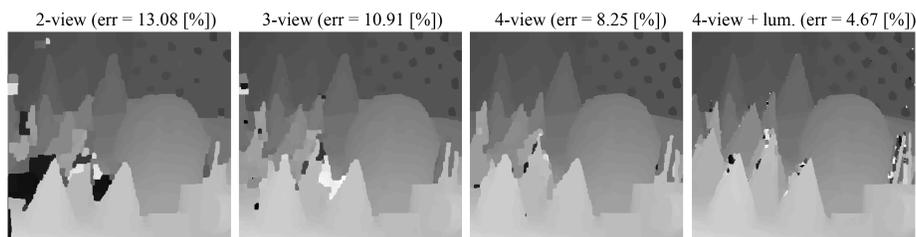


Fig. 6. Disparity maps generated using two, three and four views

Even though the disparity maps, which are computed using Census transform are more accurate, the differences in the fused images are quite imperceptible. Some of the errors in the disparity map are only slightly inaccurate. Moreover, it can be noted that even though the image fusion is based on the disparity map, the errors in the disparity map do not necessarily propagate to the fused image. For example, there are erroneous disparities in the right side of the teddy bear in Figure 5 but there are no color errors in the corresponding areas in the fused image. This is true for many other areas in all of the datasets.

On the other hand, even the ground truth disparity map does not give the perfect output image because occlusions are not considered. In fact, for all Middlebury datasets it holds that the estimated disparity map gives better results than the ground truth map. In the estimated disparity map, the occluded areas are interpolated from the occluder rather than from the occludee. From the viewpoint of the first view, this will result to somewhat incorrect disparity map. However, such disparity map works better for the image fusion.

In general, color errors are most noticeable in occluded areas and near discontinuities. This is expected because proper occlusion handling is not implemented. Figure 7a shows a smaller image patch chosen for the closer inspection (blue rectangle). The red flower on the foreground occludes some of the grass on the background. These areas are not visible in the blue filtered image. Consequently, the corresponding areas in the fused image have turned blue. The color

error results from the fact that missing color values in the blue filtered image are taken from the pixels that belong to red flower.

All tests were performed with a desktop PC that has Intel Core i5 3.20 GHz CPU and 8 GB of RAM. Computational time highly depends on the chosen disparity estimation method, image size and disparity range. Not surprisingly, the graph cut method is significantly slower than the semi-global matching. For example, the average running time of the graph cuts method with Census transform is 69 seconds for the Tea dataset and 55 seconds for the Grass dataset. The corresponding times for the semi-global matching are 8.4 s and 4.9 s.

The result of synthetic refocusing on Grass dataset is shown in Figure 7(b-c). The underlying disparity map was computed using SGM and Census transform. The overall quality of the depth of field effect is good. The refocusing ability depends on the accuracy of the disparity map. There are small inaccuracies in the disparity map near the edges of the flower (red rectangle). As a result, some of these areas are unrealistically blurred in the refocused image (yellow rectangle). Errors are most visible in the middle of the image where foreground is in focus.

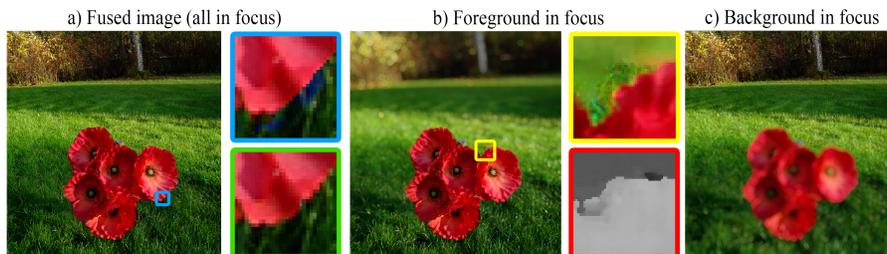


Fig. 7. Synthetic refocusing on Grass dataset. Details from the reference image (green), fused image (blue), foreground in focus image (yellow) and disparity map (red).

4 Conclusion

An image fusion algorithm was designed and implemented for a four-aperture camera. According to experiments, the semi-global matching with Census transform gave the best overall performance. The quality of the fused images is near the reference images. Closer inspection of the fused images reveals small color errors, typically found near the object borders. Future improvements, such as occlusion handling would significantly increase the quality of fused images.

It was also demonstrated that the robustness of disparity estimation increases when matching costs from multiple views are combined. Event though this work is focused on image fusion, similar approach could be used in other multi-spectral matching problems. One could also add more cameras to the system without significantly increasing the computation time. Disparity estimation would stay the same, only the matching costs would be different. Moreover, there are no limitation on how cameras are arranged since algorithm utilizes trifocal tensors. Our test setup did not show all the advantages of the actual four-aperture camera because test images were captured with a Bayer filter camera. However, the

promising test results imply that further research and development of the algorithm is desirable. The four-aperture camera has potential to become a serious competitor to the traditional Bayer matrix cameras in portable devices.

References

1. Suda, Y.: Image sensing apparatus and its control method, control program, and storage medium for correcting position deviation of images. US Patent No. 7847843 (2010)
2. Yu, Y., Zhang, Z.: Digital cameras using multiple sensors with multiple lenses. US Patent No. 6611289 (2003)
3. Kolehmainen, T., Rytivaara, M., Tokkonen, T., Mäkelä, J., Ojala, K.: Imaging device. US Patent No. 7453510 (2008)
4. Gere, D.,S.: Image capture using luminance and chrominance sensors. US Patent No. 8497897 (2013)
5. Sung, G.-Y., Park, D.-S., Lee, H.-Y., Kim, S.-S., Kim, C.-Y.: Camera module. European Patent No. 1871091 (2007)
6. LinX Imaging, Technology presentation (2014), <http://linximaging.com/imaging/>
7. Venkataraman, K., Lelescu, D., Duparre, J., McMahon, A., Molina, G., Chatterjee, P., Mullis, R., Nayar, S.: PiCam: An ultra-thin high performance monolithic camera array. In *ACM Transactions on Graphics*, Vol. 32, No. 6, 13 p. (2013)
8. Hernández, C.: Lens blur in the new google camera app (2014), googleresearch.blogspot.com/2014/04/lens-blur-in-new-google-camera-app.html
9. Kim, J., Kolmogorov, V., Zabih, R.: Visual correspondence using energy minimization and mutual information. In *The Proceedings of the 9th IEEE International Conference on Computer Vision*, Vol. 2, pp. 1033-1040 (2003)
10. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 328-341 (2008)
11. Hartley, R., Zisserman, A.: *Multiple view geometry in computer vision*. 2nd Edition, Cambridge University Press, United States of America, 655 p. (2003)
12. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 9, pp. 1582-1599 (2008)
13. Egnal, G.: Mutual information as a stereo correspondence measure. University of Pennsylvania, Department of Computer and Information Science, Technical Report No. MS-CIS-00-20 (2000)
14. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In *Lecture Notes in Computer Science*, Vol. 801, pp. 151-158 (1994)
15. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, pp. 1222-1239 (2001)
16. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, Vol. 47, No. 1-3, pp. 7-42 (2002)
17. Scharstein, D., Szeliski, R.: High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 195-202 (2003)