Noname manuscript No.
(will be inserted by the editor)

Parallax Correction via Disparity Estimation in a Multi-aperture Camera

Janne Mustaniemi · Juho Kannala · Janne Heikkilä

Received: date / Accepted: date

Abstract In this paper, an image fusion algorithm is proposed for a multi-aperture camera. Such camera is a feasible 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 single-spectral images. We improve the disparity estimation by combining matching costs over multiple views using 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 · Graph cuts · Semi-global matching

1 Introduction

A 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.

The multi-aperture camera has several advantages compared to the traditional Bayer matrix 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. This is particularly problematic in mobile devices in which low-profile cameras are needed. Multi-aperture camera solves this problem by using a combination of smaller sensors, each having dedicated optics with reduced optical height. An example of multi-aperture camera is shown in Figure 1. [3]

The image sensor measurements are subject to photon and electron leakage, which complicates the reconstruction of the desired image signal. The problem is that neighboring pixels may interact with each other. This phenomenon is known as crosstalk. It occurs when photons received by one pixel are falsely sensed by other pixels around it. The crosstalk is expected to become more severe as the image resolution continues to increase and the pixel sensors are more densely backed together. This is problematic since pixels are smaller and closer together. In Bayer filter cameras, the adjacent pixels capture the light intensity of different color bands. For such cameras, the most noticeable consequence of crosstalk is the desaturation of color [6]. The multi-aperture camera in Figure 1 does not have that problem. The camera is equipped with red, green and blue color filters meaning that each sensor is only measuring a single spectral color. Furthermore, the fourth camera captures the luminance information of the scene,

J. Mustaniemi · J. Kannala · J. Heikkilä
Center for Machine Vision Research, Department of Computer Science and Engineering, P.O Box 4500, FI-90014 University of Oulu, Finland
E-mail: janne.mustaniemi@ee.oulu.fi

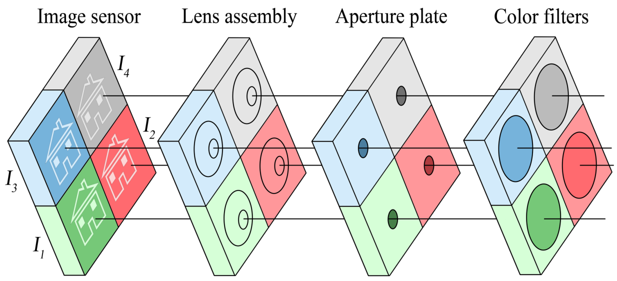

Fig. 1 Image sensing arrangement of the four-aperture camera

which can be used to increase the light sensitivity of the camera.

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 [7]. The effect can be seen as colored and blurred edges especially along boundaries that separate dark and bright parts of the image. In the multi-aperture camera, the chromatic aberration between the sensors can be avoided by calibrating them for different wavebands. It should be also noticed that the aperture of a single sensor can be smaller than in a conventional camera for achieving equivalent f-number. This increases the depth of field, which reduces the problem of varying focus distances. The lenses in the multi-aperture camera can also be simpler because the chromatic aberration is less of a problem when designing the optics. Besides the improved image quality, a simpler design usually means lower manufacturing costs. It can be noted that even though we have significantly less chromatic aberration in the green, red and blue filtered images, the luminance image may still suffer from chromatic aberration.

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 [11] 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 [8,9].

There already exist patents of multi-aperture cameras [1,2]. Some of the largest mobile phone companies have also patented their versions of the cameras [3–5]. Probably the most complete implementations of multi-aperture camera modules come from LinX Imaging [8], Pelican Imaging [9] and Light [10].

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 the technology presentation in [8], 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 a commercial 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. [9]

Most recent announcement in the field comes from a company called Light [10]. The Light L16 multi-aperture camera contains 16 individual camera units. The camera captures ten images simultaneously with different focal lengths and fuses them together. Resolution of the final image is up to 52 megapixels. The camera seems to offer a good low-light performance based on the sample images captured by the latest prototype. There is also possibility to change the depth of field and focus of the image after the image has been captured.

In this paper, we propose a novel image fusion algorithm for a four aperture camera in Figure 1. The final image is formed by combining the sub-images into a single RGB image. 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 [12,13] 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 parallax error arising from the distances of the lenses is taken into account when fusing the images. The algorithm is based on disparity estimation, in which the aim is to find corresponding pixels from each image. Disparities are estimated from the single-spectral images captured by the

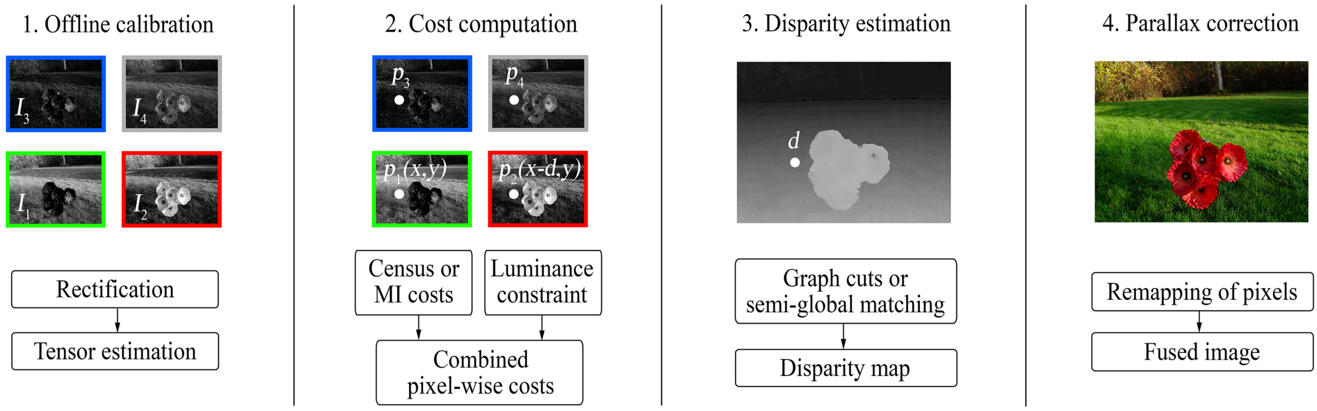

Fig. 2 Processing steps of the image fusion algorithm

four-aperture camera. Parallax error between the images is then corrected using the disparity map. The processing steps of the algorithm are shown in Figure 2.

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 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. After the rectification, the corresponding pixels are located on the same horizontal pixel rows. Other images are not rectified because algorithm utilizes trifocal tensors.

Image fusion can be performed by matching each image pair independently. However, such approach would not utilize the full potential of multiple views. Arranging cameras to have both horizontal and vertical baselines can resolve ambiguities that are common in two-view case. For example, matching a pixel that is located on the edge, parallel to baseline. Also the robustness against noise 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. It allows images to be processed together, instead of matching each image pair independently. Trifocal tensor encapsulates all the geometric relations among three views. It only depends on the motion between the views and the internal parameters of the cameras [14]. 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 matrices are assumed to be unknown and they are calibrated in this process.

In practice, one can use the tensor to transfer a 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 between the four images. 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 us assume that there is a point $p_1 = (x, y)$ in the first image and its disparity d in relation to the second image is known. Then, the corresponding points in the second, third and fourth images are computed as follows:

$$\begin{aligned} p_2 &= (x - d, y) \\ p_3 &= transferPoint(p_1, p_2, T_1) \\ p_4 &= transferPoint(p_1, p_2, T_2) \end{aligned} \tag{1}$$

It can be noted that when a point is transferred, the new coordinates are not exact integer values. In other words, the point does not correspond to any particular pixel. The corresponding pixel intensity is computed from the neighboring pixels using bilinear interpolation. For the technical details of the trasferPoint function we refer to [14, p. 382].

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. The simplest way to measure whether two pixels are similar is by taking their absolute intensity difference. This constant intensity assumption is often violated in the presence of radiometric differences such as lighting and exposure changes or noise. Similar problems arise when cameras are equipped with different color filters. This work utilizes mutual information and Census transform as similarity measures. They both are known to be robust against radiometric differences [15,16].

To further improve the robustness of disparity estimation we use a luminance cost C_L, which is combined with the mutual information or the Census transform costs. The cost of assigning disparity d for a pixel p is defined as follows:

$$C(p, d) = C_{MI/census} + K \cdot C_L, \tag{2}$$

where K is a constant, which controls the influence of the luminance cost C_L. The terms $C_{MI/census}$ and C_L are explained next.

2.2.1 Mutual Information

Mutual information (MI) has been used as a similarity measure with local [16] and global [12,13] 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:

$$MI_{I_1,I_2} = H_{I_1} + H_{I_2} - H_{I_1,I_2}, \tag{3}$$

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 the 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 [16]. 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 [12] used mutual information as a pixelwise matching cost. The computation of joint entropy H_{I_1,I_2} was transformed into a cost matrix $h_{I_1,\bar{I}_2}(i_1, i_2)$ using Taylor expansion. The cost matrix contains costs for each combination of pixel intensities $I_1(p) = i_1$ and $\bar{I}_2(p) = i_2$.

The cost matrix is computed iteratively using the full images and the disparity map from the previous iteration. A single iteration is visualized in Figure 3. Note that pixels in the second image need to be remapped $\bar{I}_2 = f_D(I_2)$ according to current disparity map D. After remapping, the corresponding pixels will have the same image coordinates (apart from the occlusions) if the disparity map is correct. The correct disparity map will also maximize the mutual information between the images. At the end of each iteration we will have a new cost matrix, from which we can estimate a new disparity map for the next iteration. The idea is that the disparity map becomes more accurate after each iteration. Disparity estimation methods are discussed in Section 2.3.

In our case, there are four images and three different cost matrices $h_{I_1,\bar{I}_2}$, $h_{I_1,\bar{I}_3}$ and $h_{I_1,\bar{I}_4}$. At the beginning of each iteration, we remap the pixels in images I_2, I_3 and I_4 with the help of current disparity map and trifocal tensors. This can be done using the Equations 1. The 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. Usually only a few number of iterations (e.g. 3 iterations) are needed until the disparity map no longer improves. The cost matrix for the image pair I_1 and I_2 is computed with formula:

$$h_{I_1,\bar{I}_2}(i_1, i_2) = -\frac{1}{n} log((P_{I_1,\bar{I}_2}(i_1, i_2) * g(i_1, i_2)) * g(i_1, i_2) \tag{4}$$

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.

The final mutual information matching cost is a combination of three cost matrices. The cost of assigning disparity d for a pixel p is defined as follows:

$$C_{MI}(p, d) = h_{I_1,\bar{I}_2}(i_1, i_2) + h_{I_1,\bar{I}_3}(i_1, i_3) + h_{I_1,\bar{I}_4}(i_1, i_4). \tag{5}$$

Even though the previous equation only contains three cost matrices, we could also compute similar matrices between all image pairs. In our experiments, we found

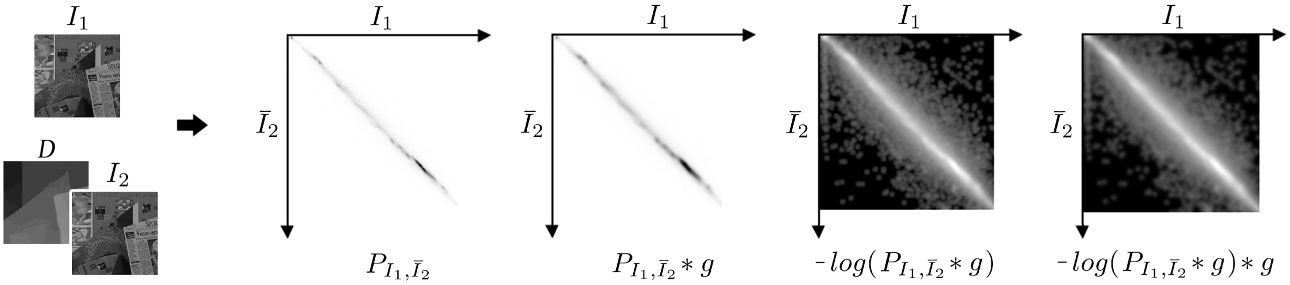

Fig. 3 Computation of mutual information cost matrix $h_{I_1,\bar{I}_2}$

that such approach did not improve the results noticeably. Therefore, we concluded that the above approach is sufficient.

2.2.2 Census transform

Census transform is based on the relative ordering of local intensity values. It can tolerate all radiometric distortions that preserve this ordering [17]. 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:

$$R_p = \bigotimes_{[x,y]\in W} \xi(p, p + [x, y]), \tag{6}$$

where symbol $\otimes$ denotes concatenation and W is the window around pixel p. The comparison operation $\xi(p, p+[x, y])$ equals to 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 since it gave good results in practise. Each pixel inside the window is compared to the center pixel. This will result to a bit string that consists of 62 bits and it can be conveniently stored into a 64 bit integer. The above 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:

$$\begin{aligned} C_{census}(p, d) = H(R_{p,1}, R_{p,2}) + H(R_{p,1}, R_{p,3}) \\ +H(R_{p,1}, R_{p,4}). \end{aligned} \tag{7}$$

2.2.3 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:

$$\hat{I}_4(p_4) = G \cdot I_1(p_1) + R \cdot I_2(p_2) + B \cdot I_3(p_3), \tag{8}$$

where point's intensity is denoted by $I(p)$. The coefficients G, R and B depend on the color filters of the cameras. They should be defined via photometric calibration to match the properties of the color filters. In case there is a large difference between the left and right side of the above equation, it is likely that points are not correspondences. Based on this assumption, the luminance cost can be written as:

$$C_L = |I_4(p_4) - \hat{I}_4(p_4)|. \tag{9}$$

In practice, one might find that Equation 8 does not perfectly hold for all frequencies in the visible light. In such case, the luminance cost could be modified to use more robust cost, e.g. Census transform. Our experiments did not show noticeable improvement when using Census transform instead of Equation 9 directly.

2.3 Disparity Estimation

Disparity estimation methods aim to find correct disparities for every pixel in the image based on the matching costs. We evaluate two different methods, graph cuts and semi-global matching. As already mentioned, here the disparity refers to the horizontal coordinate difference between the corresponding pixels in the first and second image. These disparities relate to other images via trifocal tensors according to Equations 1.

For each pixel in the first image, we need to consider all possible disparities within a given disparity range. The simplest way would be to choose the disparity value that minimizes the matching cost between the pixels. It is common that incorrectly matched pixel gets a lower matching cost because of the image noise, lightning variations, occlusions etc. Graph cuts and semi-global matching also consider the smoothness of the disparity map. They use the assumption that in the real world, the changes in the scene will vary smoothly and therefore, the neighboring pixels can be assumed to have similar disparities.

Many of the computer vision problems can be expressed in terms of energy minimization. In case of disparity estimation, the goal is to assign disparity values for each pixel in such way that global energy function is minimized. Graph cuts and semi-global matching are both based on energy minimization. These methods are well-known and relatively well placed in the Middlebury stereo evaluation website [26]. This is the main reason why we decided to use them in this work.

2.3.1 Graph Cuts

Instead of computing disparities for each pixel independently, graph cuts method performs a global optimization process over the whole image. The idea is to construct a specialized graph for the energy function. The energy is minimized with a max flow algorithm that finds the minimum cut on the graph. A general form of the energy function is:

$$E(D) = \sum_{p} C(p, D_p) + \lambda \sum_{p,q \in N} V_{p,q}(D_p, D_q), \tag{10}$$

where, the first term is the sum of all matching costs when using the disparity map D. The second term is the smoothness term. The set of pairs of adjacent pixels is denoted by N. The neighborhood interaction function $V_{p,q}(D_p, D_q)$ assigns higher penalties for pairs of neighboring pixels if they have different disparities. Since disparity can also change rapidly at the object boundaries, this should be a robust function, which can preserve discontinuities. Scale factor λ controls the influence of the smoothness term.

We employ the multi-label optimization library developed by Veksler et al. [18,19,21]. With this library, either expansion move or swap move algorithm can be used to minimize the global energy function. The expansion move algorithm was chosen since it gave slightly better results and was faster than the swap move algorithm. After testing different smoothness costs with varying parameters, truncated absolute distance was chosen. It gave the best overall performance compared to Potts model and truncated quadratic difference.

2.3.2 Semi-global Matching

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. The energy is defined by the formula:

$$\begin{aligned} E(D) = \sum_{p} (C(p, D_p) + &\sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] \\ &+ \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]). \end{aligned} \tag{11}$$

The first term is the sum of all matching costs when disparity map D is used. The latter terms penalize the disparity differences of neighboring pixels N_p with the costs P_1 and P_2. The larger cost P_2 is added when the disparity differs more than one pixel.

The energy function is computed with dynamic programming along 1D paths from 8 directions towards each pixel of interest. The costs of all paths are then summed and the final disparity is determined by winner-takes-all approach.

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 presented in [13]. 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 images I_3. 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. The maximum disparity depends on the baseline of the cameras and the distance between the camera and the closest object in the scene.

Now that corresponding pixels have the same image coordinates, an RGB image can be constructed by simply combining images I_1, I_2 and I_3. In our implementation, the luminance image I_4 is not used when forming the final image. However, the fourth image could be used to improve the signal-to-noise ratio of the luminance channel. This would be particularly useful in low-light conditions.

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. In each camera position, one of the channels was chosen. Luminance image was created from the original RGB image by weighting each color component with coefficients $G = 0.587$, $R = 0.299$ and $B = 0.114$. We used these same coefficients in the Equation 8.

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 all cameras are parallel to each other [22, 23]. Ground truth disparity maps were only 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.

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. Disparity was classified as invalid if its value differs more than 1 pixel from the ground truth. Disparities were not evaluated in occluded areas since occlusion handling was not implemented.

Table 1 Image sizes and disparity ranges in pixels

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

Smoothness parameters of the semi-global matching and graph cuts methods were manually tuned for the mutual information and Census transform costs. 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. Figure 5 shows the result of image fusion for Tea dataset. The image was created by using graph cuts with mutual information. In comparison to original RGB image in the same figure, it can be concluded that visual quality of the fused image is near the reference image. In contrast, the right most image in Figure 5 shows the output obtained without correcting the parallax error. The image is constructed by simply combining the red, green and blue channels of the original images. As can be seen, the resulting image has very severe color errors.

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 6.

The advantages of using trifocal tensor and four different views are best demonstrated with disparity maps. The left most disparity map in Figure 7 is generated

Table 2 Results of graph cuts method

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

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

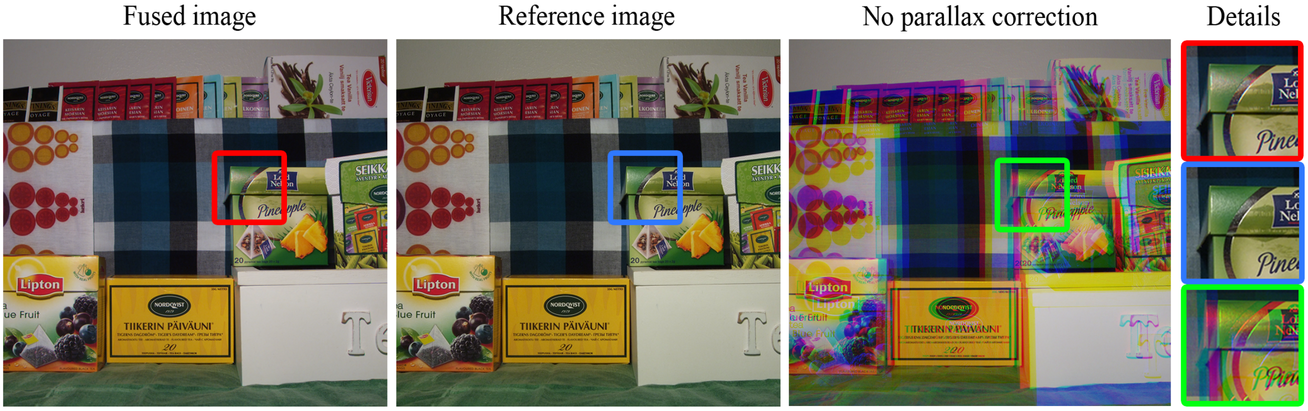

Fig. 5 The result of graph cuts with mutual information on Tea dataset. For demonstration, we also show the result obtained without parallax correction

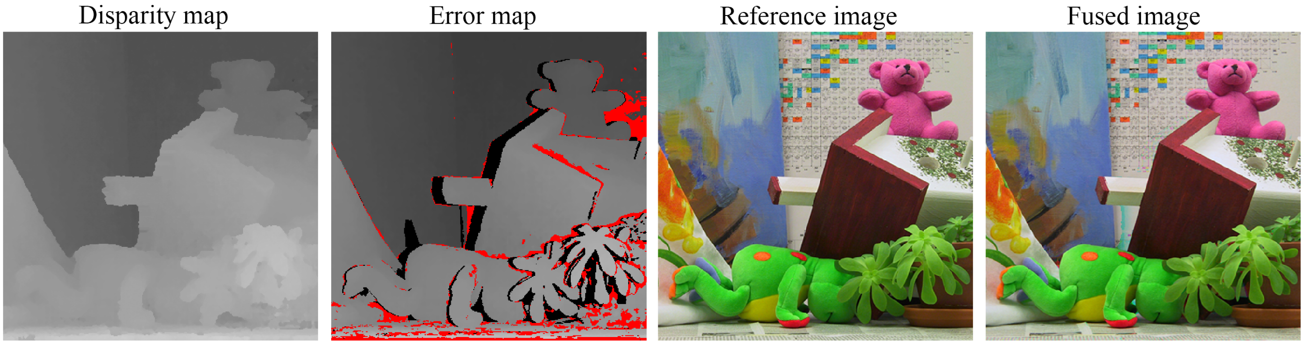

Fig. 6 The result of semi-global matching and Census transform on Teddy dataset. Red areas in the error map represent erroneous disparities and black areas are occlusions

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 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.

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 6 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. Generally, the errors are not visible if the erroneous disparities are located in non-textured areas.

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 interpo-

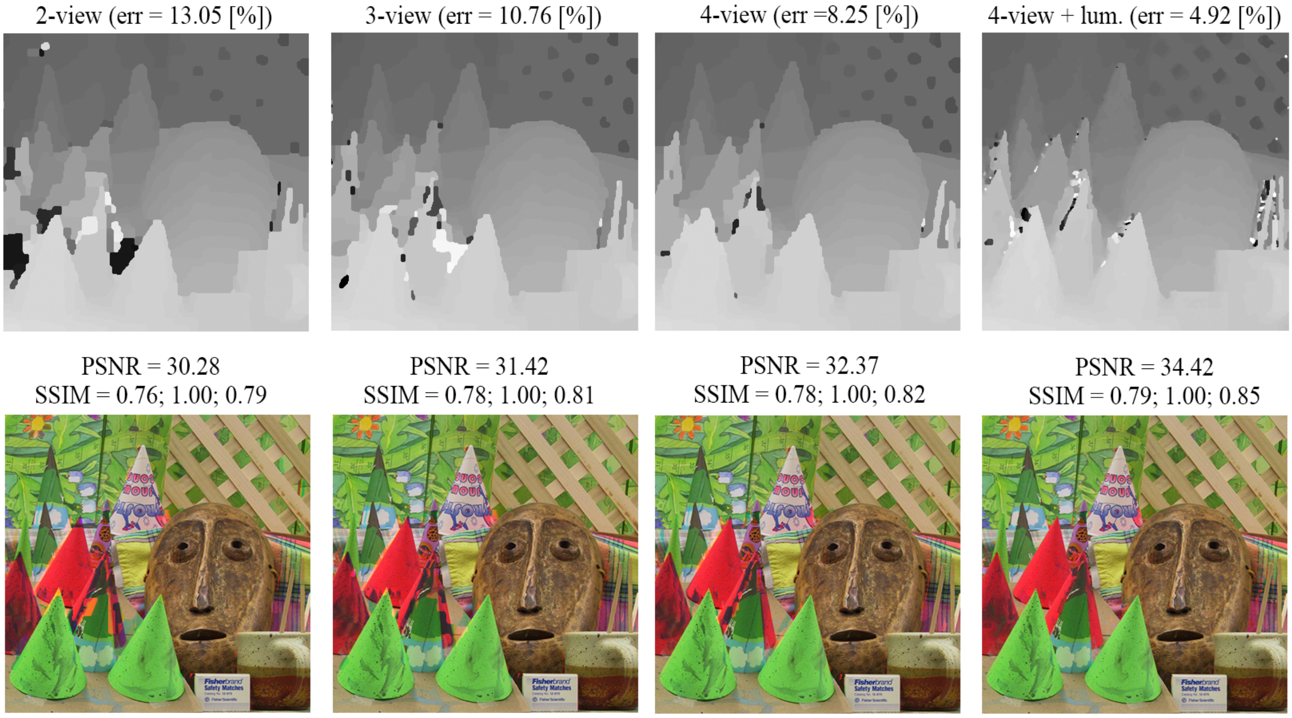

Fig. 7 Disparity maps generated using two, three and four views. Bottom row shows the corresponding fused images.

lated 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. Blue rectangle in Figure 9 shows a smaller image patch chosen for the closer inspection. 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 the red flower. One can see similar problems in the Flower dataset in Figure 8. The pink rectangle reveals that the background has turned from white to yellow. It can be noted that this type of color errors can also be more noticeable depending on the foreground and background colors. For example, if in the previous case, the flower was white and the background was red, then we could expect magenta like errors.

Table 3 Results of semi-global matching

	Mutual Information		
Dataset	Errors	PSNR	SSIM (rgb)
Teddy	10.92	38.43	0.88; 1.00; 0.81
Cones	6.84	34.95	0.86; 1.00; 0.79
Venus	2.96	41.22	0.91; 1.00; 0.83
Tea	-	39.47	0.95; 1.00; 0.88
Flowers	-	40.05	0.94; 1.00; 0.87
Grass	-	34.19	0.82; 1.00; 0.84
	Census Transform		
Dataset	Errors	PSNR	SSIM (rgb)
Teddy	6.81	38.32	0.89; 1.00; 0.81
Cones	4.67	35.10	0.87; 1.00; 0.79
Venus	1.30	40.40	0.90; 1.00; 0.83
Tea	-	40.36	0.96; 1.00; 0.89
Flowers	-	40.12	0.95; 1.00; 0.87
Grass	-	35.01	0.86; 1.00; 0.87

The red rectangle in Figure 5 shows details of the Tea dataset. It can be seen that the gray strip in pineapple tea box has slightly changed its color in compared to reference image. In fact, the strip has a chrome coating, which makes it extremely reflective. Errors caused by the reflections are difficult to avoid completely since even the correct disparities may result to unexpected color artifacts.

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

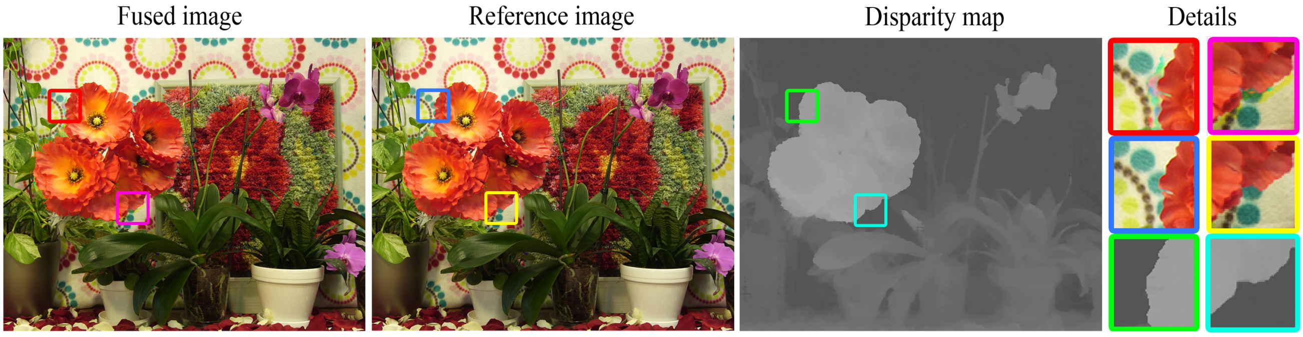

Fig. 8 The result of semi-global matching with Census transform on Flower dataset

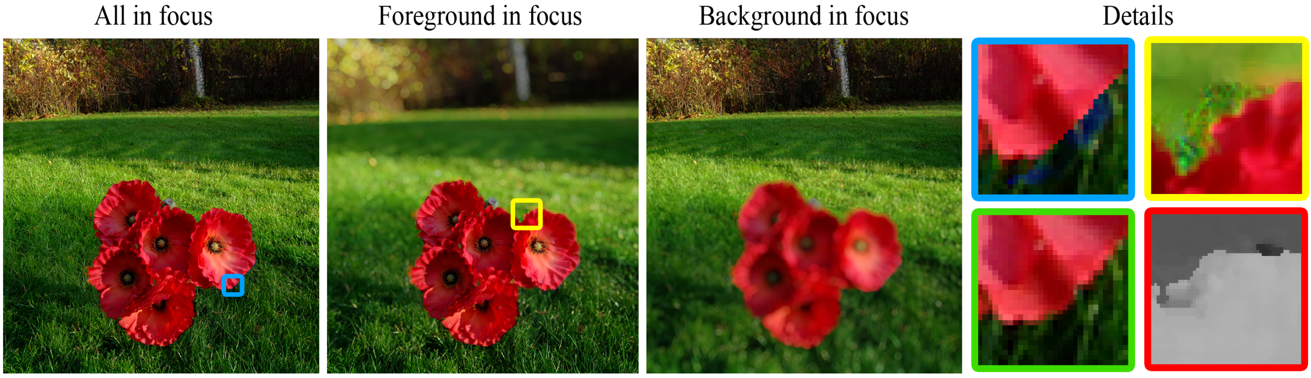

Fig. 9 Synthetic refocusing on Grass dataset. Details from the reference image (green), fused image (blue), foreground in focus image (yellow) and disparity map (red)

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 9. 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.

3.1 Occlusion handling

Experiments showed that color errors are typically found near the object borders. This is mainly caused by the fact that our implementation does not consider occlusions. A proper occlusion handling would significantly increase the quality of fused and refocused images. In this paper, we did not address this problem because the effects of occlusions depend much on the camera setup and configuration.

In our experiments, the baseline was relatively large and the objects were close to the camera. This emphasizes the need for occlusion handling. We wanted to make sure that possible matching errors would be clearly visible and that occlusion problem was also covered although not solved. In typical shooting situations, we would not expect the occlusions to be as severe. This is the case, especially if the baseline was smaller. Nonetheless, the occlusion handling would be an important direction for future research.

The occlusion handling consist of two main steps: occlusion detection and occlusion filling. In the first step, we would detect the areas in the reference image (green image), which are not visible in the red and blue filtered images. In the second step, we would fill the missing color values using inpainting or colorization methods.

A simple way to detect occlusion and false matches is to perform disparity calculation in both directions. That is, matching the first image to the second and then the other way around. This will produce two slightly different disparity maps in which the inconsistent disparities represent occlusions or false matches. Alternatively, it is possible to detect occlusions by encoding the visibility constraint directly into the global energy function [20].

After the occlusion detection, we need to assign values for the occluded pixels. The missing color values would be interpolated from the neighborhood. More specifically from the background rather than from the foreground object. Here we could also utilize colorization methods such as [24].

4 Conclusion

In this paper, 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. Even though this work is focused on the 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.

It is safe to say that our implementation does not meet the time constrains of a real four-aperture camera. However, there exists fast GPU based implementations of semi-global matching. For instance, the algorithm presented in [25] could easily solve the disparities in a fraction of a second even if we would use larger input images and greater disparity range.

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. On the other hand, this was not a problem since evaluation was more focused on matching performance rather than improvements in image quality. After all, the disparity estimation plays a very important role what it comes to the quality of the final image. The promising test results imply that further research and development of the algorithm is worthwhile. 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. Hirakawa, K.: Cross-talk explained. 15th IEEE International Conference on Image Processing, pp. 677-680 (2008)
7. van Walree, P.: Chromatic aberrations. http://toothwalker.org/optics/chromatic.html. Accessed 5 Apr 2016
8. LinX Imaging, Technology presentation (2014), http://linximaging.com/imaging/
9. 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)
10. Light, https://light.co/camera. Accessed 25 November 2015
11. Hernández, C.: Lens blur in the new google camera app (2014), googleresearch.blogspot.com/2014/04/lens-blur-in-new-google-camera-app.html
12. 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)
13. 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)
14. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. 2nd Edition, Cambridge University Press, United States of America, 655 p. (2003)
15. 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)
16. 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)
17. Zabih, R., Woodfill, J.: Non-parametric local transforms for computing visual correspondence. In Lecture Notes in Computer Science, Vol. 801, pp. 151-158 (1994)
18. 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)
19. Kolmogorov, V., Zabih, R.: What Energy Functions can be Minimized via Graph Cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 2, pp. 147-159 (2004)
20. Kolmogorov, V., Zabih, R., Gortler, S.: Generalized Multi-camera Scene Reconstruction Using Graph Cuts. Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, Vol. 2683, pp. 501-516 (2003)
21. Boykov, Y., Kolmogorov, V.: An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Mini-

mization in Vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 9, pp. 1124-1137 (2004)

22. 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)
23. 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)
24. Levin, A., Lischinski, D., Weiss, Y.: Colorization Using Optimization. ACM SIGGRAPH 2004, pp. 689-694 (2004)
25. Banz, C., Blume, H., Pirsch, P.: Real-time semi-global matching disparity estimation on the GPU. In IEEE International Conference on Computer Vision Workshops, pp. 514-521 (2011)
26. Middlebury Stereo Evaluation. http://vision.middlebury.edu/stereo/eval/. Accessed 5 Apr 2016