J Supercomput (2018) 74:3666–3684
https://doi.org/10.1007/s11227-016-1884-7

Fast depth estimation using semi-global matching and adaptive stripe-based optimization

Maziar Loghman[1] · **Joohee Kim**[2] · **Kyuwon Choi**[2]

Published online: 17 October 2016

Abstract Depth estimation is one of the essential components in many computer vision applications such as 3D scene reconstruction and stereo-based object detection. In such applications, the overall quality of the system highly depends on the quality of depth maps. Recently, several depth estimation methods based on semi-global matching (SGM) have been proposed because SGM provides a good trade-off between runtime and accuracy in depth estimation. In addition, non-parametric matching costs have drawn a lot of attention because they tolerate all radiometric distortions. In this paper, a depth estimation method based on the Census transform with adaptive window patterns and stripe-based optimization has been proposed. For finding the most accurate depth value, adaptive length optimization paths via multiple stripes are used. A modified cross-based cost aggregation technique is proposed which adaptively creates the shape of the cross for each pixel distinctly. In addition, a depth refinement algorithm is proposed which fills the holes of the estimated depth map using the surrounding background depth pixels and sharpens the object boundaries by applying a trilateral filter to the generated depth map. The trilateral filter uses the curvature of pixels as well as texture and depth information to sharpen the edges. Simulation results indicate that the proposed methods enhance the quality of depth maps while reducing

✉ Joohee Kim
joohee@ece.iit.edu

Maziar Loghman
mloghma1@hawk.iit.edu

Kyuwon Choi
kchoi@ece.iit.edu

[1] Multimedia Communications Laboratory, Illinois Institute of Technology, Chicago, IL, USA

[2] Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL, USA

the computational complexity compared to the existing SGM-based depth estimation methods.

Keywords Depth estimation · Cross-based cost aggregation · Stripe-based optimization · Advanced driver assistance system

1 Introduction

Depth is considered as one of the most important cues in perceiving the three-dimensional characteristics of objects in the scene captured by cameras. In computer vision, the value which represents the distance between each object in the scene to the focal point of the camera is called depth and an image storing these values for all the pixels is referred to as a depth map. Depth maps are essential in a variety of applications such as view synthesis, robot vision, 3D scene reconstruction, interaction between human and computer, and advanced driver assistance systems [1–3]. The performance of the mentioned applications is highly dependent on the quality and accuracy of the depth map. Thus, generating an accurate depth map is of substantial importance. The main objective of depth estimation methods is to generate a per-pixel depth map of the scene based on two or more reference images of the same scene from different angles. The reference images are captured by a stereo calibrated camera system in which the cameras are parallel to each other or are set with a slight angle. The images are typically stereo rectified with the property that the corresponding points in each image have identical vertical coordinates.

Most of stereo matching methods usually consist of the following four steps:

- *Matching cost calculation* The similarity of image locations is measured by defining a matching cost. Normally, a matching cost is calculated at each pixel for all disparities under certain considerations. Common matching costs include absolute differences (AD) and squared differences (SD). More complicated matching costs have also been proposed such as mutual information (MI), Census transform, and rank transform. An evaluation of different cost functions for stereo matching is provided in [4].
- *Cost aggregation* The calculated matching cost is aggregated over a support region. Numerous cost aggregation methods have been proposed recently [33–35]. The most common way is by using a square window. Cross-based cost aggregation is an alternative solution to this matter.
- *Disparity optimization* Disparity refers to the distance between two corresponding points in the left and right images of a stereo pair. The disparity which minimizes the cost function is chosen as the optimum value and the depth value is calculated accordingly. Several optimization techniques have been proposed to estimate the optimum depth value.
- *Depth refinement* Depending on the application, the requirement for depth map accuracy varies. Some applications like human pose estimation in the gaming industry are satisfied by low resolution depth maps. However, higher accuracy is vital for applications like depth image-based rendering (DIBR) techniques and stereo-based advanced driver assistant systems (ADAS).

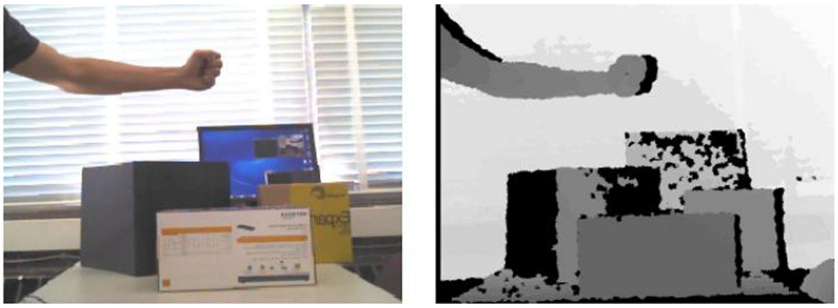


Fig. 1 Example of a Kinect dataset **a** color image, and **b** depth map (color figure online)

Depth maps can be estimated using either stereo matching techniques or depth sensors. With the advent of depth sensors, novel camera systems have been developed [5], which generate depth maps in real-time. The measurement of depth in such sensors is often performed by either using time-of-flight (TOF) systems or infrared pattern deformation. Depth maps acquired by the depth sensors are usually noisy and suffer from poorly generated depth boundaries. Figure 1 shows an example of a set of color and depth map of a scene captured by Kinect device [5]. We can see that the depth map has extremely poor quality.

Stereo matching techniques can be classified into two groups, namely local and global techniques. Local methods [6,7] consider a finite-size window to estimate the disparity. Thus, the window size plays an important role in such methods. The local methods are fast and computationally simple but they are highly error-prone and the estimated depth maps are usually inaccurate. On the other hand, in global techniques an energy function is globally optimized to find the disparity. Global depth estimation techniques can generate high-quality depth maps. Most popular techniques in this category include belief propagation [8], graph cuts [9] and dynamic programming [10]. However, due to the computational complexity of such algorithms, it is not feasible to use them in real-time applications. Semi-global matching (SGM) [11] which was first introduced by Hirschmuller, performs pixelwise matching based on mutual information and approximation of a global smoothness constraint. Therefore, a good trade-off between accuracy and runtime is obtained. However, it achieves limited performance under illumination changes. Despite the presence of such promising depth estimation techniques, there are still several problems in the generated depth maps. The existence of holes and sensitivity to noise and illumination changes are the main significant problems. These problems are partially solved by post-processing the depth maps using depth refinement techniques [12,13].

In this paper, we propose a depth estimation algorithm based on the Census transform and stripe-based optimization. The main contribution of the proposed method is to improve the quality of estimated depth maps with low computational complexity so that it can satisfy the requirements of real-time applications such as 3D scene reconstruction and stereo-based pedestrian detection. We focus on applying the proposed depth estimation method for stereo-based pedestrian detection in advanced driver assistance systems. The proposed method increases the performance of ADAS systems by improving the pedestrian detection rate. Simulation results indicate a significant increase in the depth map accuracy and quality. Performance evaluations are performed using standard benchmarks and real-world scenarios to demonstrate the efficiency of the proposed algorithm.

The rest of the paper is organized as follows. In Sect. 2, we review previous work related to real-time depth estimation. The proposed method is presented in detail in Sect. 3. The experimental results are provided in Sect. 4, and the paper is concluded in Sect. 5.

2 Related work

Several approaches have been proposed in the past for estimating dense disparity maps from stereo images captured by static cameras. Due to the development of various applications that require real-time processing, such as advanced driver assistance systems, real-time depth estimation has become a more attractive research area in recent years. In this section, we will briefly review recently proposed real-time depth estimation techniques.

During the past several years, many stereo matching techniques have been proposed. A survey study of different stereo matching algorithms is available in [14]. A multi-resolution stereo matching technique implemented in graphics hardware is presented in [15]. The multi-resolution scheme reduces the noise and sum-of-squared-difference (SSD) is used to aggregate matching cost. The optimum disparity values are determined based on the winner-take-all criterion. A depth estimation method for uncalibrated stereo images is proposed in [16].

Due to high computational complexity of some of stereo matching techniques, real-time performance is only achieved via specially designed hardware. In [17], a graphics processing unit (GPU)-based depth estimation method is proposed which exploits adaptive cost aggregation windows. The windows change shape according to the local content of the image, such as edges and corners. A near-real-time stereo matching technique based on dynamic programming is presented in [18]. Two different implementations are presented in the paper. The first one uses the GPU for matching cost calculation and uses the CPU for disparity estimation, while the second one uses GPU for both calculations. A real-time correlation-based stereo matching algorithm is proposed in [19] which uses an image-gradient-guided cost aggregation scheme. The scheme is designed to fit the architecture of GPUs and the whole algorithm is run on the graphics board.

Depth estimation using SGM has been studied intensively for the past years. The original SGM implementation on GPU was first presented in [20]. An implementation of SGM with 8 accumulation paths on field programmable gate array (FPGA) is presented in [21].

The method proposed in [22] uses a 4 path cost aggregation method and an extended work with fewer aggregation paths is presented in [23].

Eliminating the diagonal integration paths is an idea proposed in [24]. By only using 4 directions for cost accumulation, the method proposed in [24] achieves a run time of 19 ms for a 512×383 image resolution. iSGM is a technique introduced in [25] as a new cost integration concept for semi-global matching by iteratively reducing the search space. A GPU-based SGM depth estimation method is proposed in [26] which can achieve real-time processing on high resolution reference images. The method runs with the frame rate of 25 fps on images with 1024×768 resolution and 128 disparity levels. In [27], a two-dimensional parallelization scheme for SGM is proposed. The FPGA implementation of the algorithm achieves real-time processing for VGA image resolutions.

In the past years, a number of depth estimation methods based on the Census transform have been proposed. In [28], a SGM-based technique is proposed which uses a 5×5 Census transform for computing the similarity while conducting cost accumulation over 8 paths. In [29], a disparity map is estimated by applying the Census transform prior to the SGM-based optimization. The depth map is refined by a segmentation and plane fitting approach. The authors in [30] introduce an algorithm based on semi-global matching and the Census transform. Two sub-pixel interpolation functions are implemented to increase the accuracy at the sub-pixel level. A modified Census transform using semi-global matching is a method introduced in [31]. The authors in [32] propose an algorithm using adaptive window patterns for Census transform.

Different cost aggregation methods have been studied for the past years. In [33], a segmentation-based adaptive support region for cost aggregation is proposed. A geodesic support weight for stereo matching is proposed in [34] where the pixels with low geodesic distance are given high support weight. A cross-based cost aggregation method is proposed in [35].

Apart from existence of holes in the estimated depth maps, other artifacts may limit the efficiency of stereo matching techniques. A few limitations of the existing depth estimation methods can be listed as:

1. Sensitivity to noise and illumination changes,
2. Inaccurate object boundaries, and
3. High computational complexity.

Taking all these artifacts into consideration, the main goal of the proposed depth estimation method is to provide a high quality depth map which can be used in real-time applications. Benefiting from multiple symmetric Census window patterns and performing most of the tasks on the low-resolution images reduce the overall computational complexity and make the algorithm applicable for real-time advanced driver assistance systems.

3 Proposed method

In our proposed method, we use a stereo image pair as the reference to generate a depth map. The Census transform is chosen as the matching metric and the local characteristics of depth maps are exploited using different window patterns in Census transform. The proposed technique consists of four steps: down-sampling and mask generation, cost calculation and aggregation, semi-global optimization, and depth refinement. Figure 2 shows the block diagram of the proposed algorithm.

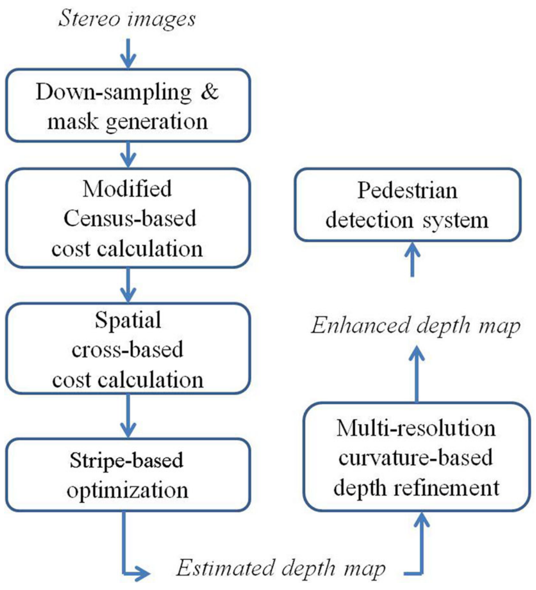

Stereo images
Down-sampling & mask generation
Modified Census-based cost calculation
Spatial cross-based cost calculation
Stripe-based optimization
Estimated depth map
Multi-resolution curvature-based depth refinement
Enhanced depth map
Pedestrian detection system

Fig. 2 Block diagram of the proposed technique

3.1 Down-sampling and mask generation

In an ideal case, pixels belonging to the same object which are positioned at the same distance from the camera should have the same depth value. However, this does not occur in real-world scenarios due to several reasons such as illumination changes within an object and mismatches in stereo matching algorithms. We use the curvature of the pixels in the given color images to distinguish between the smooth area and sharp edges of the objects in the scene. To reduce the computational complexity of the algorithm, the first step is to down sample the stereo color images by a factor of 4. Then, we obtain a mask which indicates the smoothness of different regions in the reference image.

The curvature is calculated using the first- and second-order gradients of each pixel given by Eq. (1).

$$k\left(x, y\right) = \frac{\mu_{xx}\mu_x^2 - 2\mu_x\mu_{xy}\mu_y + \mu_{yy}\mu_y^2}{\left(\mu_x^2 + \mu_y^2\right)^{3/2}}, \tag{1}$$

where μ_x and μ_{xx} are the first- and second-order gradients, respectively. Subscripts indicate the direction of gradient. The Prewitt kernel is used to find the gradient. After computing the curvature, we aggregate the values over a 5×5 window and store it in a curvature map. A binary mask is generated by Eq. (3) using the curvature map. When the aggregated curvature of a pixel is less than a threshold, a zero value is assigned to the mask. An example of mask generation is shown in Fig. 3.

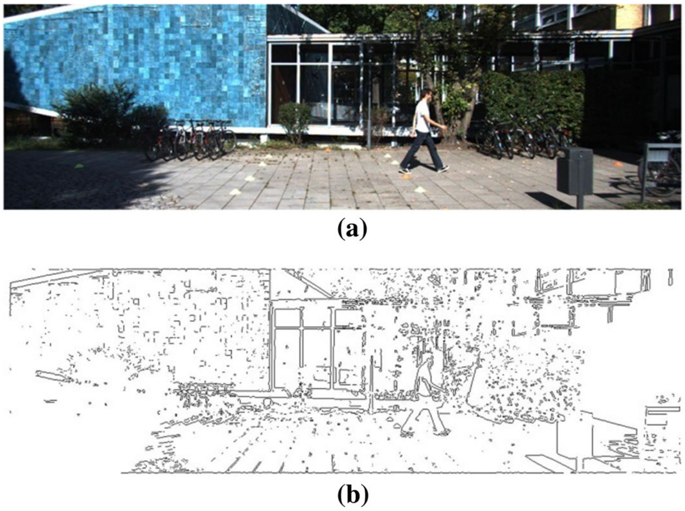


Fig. 3 Mask generation: **a** color image and **b** mask (color figure online)

$$\text{agg}_c = \sum_{\Delta y=-2}^{2} \sum_{\Delta x=-2}^{2} k\left(x + \Delta x, y + \Delta y\right), \tag{2}$$

$$M\left(x, y\right) = \begin{cases} 0, & \text{if } \text{agg}_c < T_1, \\ 1, & \text{Otherwise.} \end{cases} \tag{3}$$

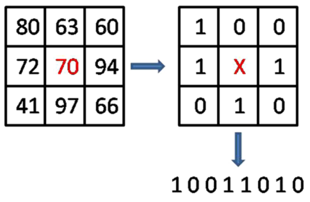
3.2 Cost calculation and aggregation

The Census transform [17] maps the local neighborhood surrounding a pixel $I_c\left(x, y\right)$ to a bit-pattern. The transform relies on the relative ordering of local intensity values and not the pixel values itself. Figure 4 shows an example of the Census transform of a window image with respect to the center pixel.

The Census transform converts relative intensity difference to 0 or 1 in a bit-pattern.

In this paper, Census transform is used to calculate the cost function for both left and right images. We use a simple Census window pattern for the smooth regions to reduce the computational complexity and use a more complex pattern for the non-uniform regions which usually contain edges and object boundaries. The adaptive Census window patterns are shown in Fig. 5 where the selected positions are denoted by black pixels.

Fig. 4 Census transform example of a window image



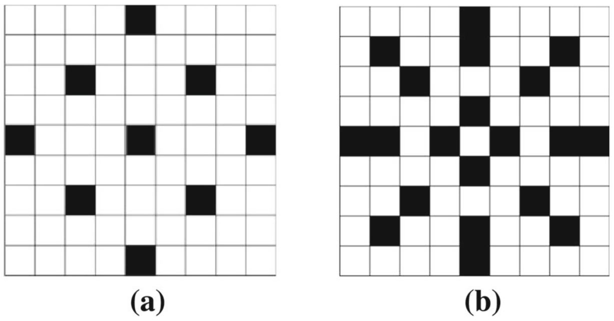


Fig. 5 Census window patterns: **a** P_1 for uniform regions, and **b** P_2 non-uniform regions

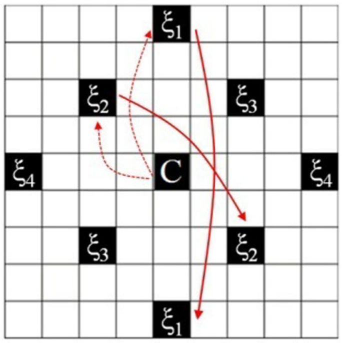


Fig. 6 Center-symmetric Census window

The black pixels shown in Fig. 5 are used in a symmetric way to generate the binary bit-pattern. Therefore, only half of the pixels are directly engaged in the computation process. The other half of pixel values are symmetrically copied using the pre-calculated data. Figure 6 shows the center-symmetric window pattern of P_1. Similarly, the binary bit-pattern of P_2 pattern is symmetrically obtained. For the pixel $I_c(x, y)$, the Census transform is calculated using Eq. (4)

$$R_T(x, y) = \otimes_{(i,j)\in N}\xi(I_c(x, y), I_c(x + i, y + j), \tag{4}$$

where N is the neighborhood of the current pixel within the Census transform, ξ is the step function and is bitwise concatenation. The step function is defined by the Eq. (5).

$$\xi(I_c(x, y), I_c(x + i, y + j) = \begin{cases} 0, & \text{if } I_c(x, y) < I_c(x + i, y + j), \\ 1, & \text{Otherwise.} \end{cases} \tag{5}$$

The binary mask generated in the previous step is used to decide which pattern to use. The decision criterion is made as follows:

$$\alpha = \sum_{(i,j)\in N} M\left(x+i, y+j\right), \tag{6}$$

$$\text{Pattern} = \begin{cases} P_1, & \text{if } \alpha < T_2, \\ P_2, & \text{Otherwise.} \end{cases} \tag{7}$$

The cost function is calculated by finding the Hamming distance between the obtained bit-patterns of the left and right reference images using Eq. (8).

$$C((x,y),d) = \sum_{(i,j)\in N} d_{\mathrm{H}}(R_{T,l}(x+i, y+j), R_{T,r}(x+i-d, y+j)) \tag{8}$$

In Eq. (8), R_T is the calculated bit-pattern. d is the disparity and d_{H} is the Hamming distance function and the subscripts l and r refer to the left and right reference images, respectively.

Since we have calculated the cost for each pixel, it is time to aggregate each pixel's cost over a support region. The main goal of cost aggregation is to reduce the matching ambiguities and noise present in the initial cost. A modified cross-based cost aggregation is proposed based on the following fact. An effective assumption is that neighboring pixels with similar colors and spatial characteristics usually belong to the same object and should have similar depth values. The proposed cost aggregation method consists of two steps:

1. Creating the cross shape, and
2. Aggregating cost over the created cross.

In the first step, an adaptive cross with varying arms is constructed for each pixel. Given a pixel p, the endpoint of the arm is defined as p_1 when one of the three following rules is not met:

– The color difference between p and p_1 should be less than a predefined threshold.
– The spatial distance between p and p_1 should be less than a preset maximum length.
– The curvature values of p and p_1 in the curvature map should not exceed a threshold.

The above-mentioned criteria are defined by Eq. (9).

$$\begin{aligned} \left|I_c\left(p\right) - I_c\left(p_1\right)\right| &\le \tau_1, \\ \parallel p - p_1 \parallel &\le L, \\ \left|\text{curv}\left(p\right) - \text{curve}\left(p_1\right)\right| &\le \tau_2, \end{aligned} \tag{9}$$

where L is the maximum length. τ_1 and τ_2 are predefined thresholds. Large thresholds are usually set for textureless regions to include adequate intensity variation.

The second step is aggregating the cost values over the created cross. The intermediate cost is obtained by summing the cost values horizontally and the final cost is calculated by adding all the intermediate data vertically. The whole process is shown in Fig. 7.

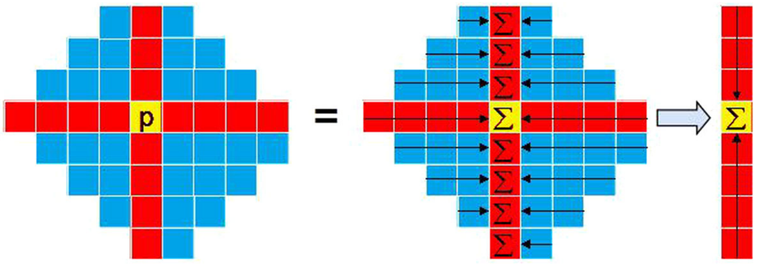
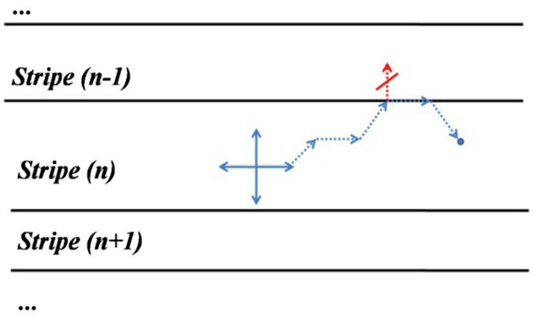


Fig. 7 Cross-based aggregation

Fig. 8 Stripe-based optimization path



3.3 Stripe-based optimization

The optimization path for finding the best match is formed within multiple-size horizontal stripes. Figure 8 shows the corresponding adaptive stripe pattern. Depending on the structure of the reference image, typically up to 10 stripes with adaptive widths are used. For each pixel, we consider 4 directions of up, down, left and right and the direction which has the least cost value is decided as the next path. The path accumulation is performed in the decided directions. However, the paths which cross the stripes are cut so that the information cannot propagate over them. The size of each stripe is obtained based on the reference image structure.

At this stage of the algorithm, we obtain the best match for each pixel and calculate the disparity. The optimal disparity is calculated by minimizing the energy function. As shown in Eq. (10), the energy function consists of three terms. First term is the matching cost from the previous step which is based on the Census transform. The other two terms are the smoothness constraints. We add two penalty terms to the matching cost function to take into account slight and abrupt changes in the disparity of neighboring pixels.

$$E\left(p, D\right) = \sum C\left(p, d_p\right) + \sum_{q \in N} f_1 F\left[|d_q - d_p| = 1\right] + \sum_{q \in N} f_2 F\left[|d_q - d_p| > 1\right] \tag{10}$$

where N is the neighborhood of the current pixel p. d_p and d_q are the depth values for pixels p and q f_1 is the penalty term when the disparity values of neighboring pixels differ by one. A larger penalty term f_2 is added when the neighboring disparity values differ by more than one.

The disparity image is estimated by finding the optimum value which minimizes the cost function as stated in Eq. (11).

$$\text{Disparity} = \text{Argmin}_d\, E\,(p, d) \tag{11}$$

The 4-direction optimization path is used to find the optimal value.

3.4 Depth refinement

The depth map obtained by the proposed modified Census transform and stripe-based optimization is further refined using the proposed depth refinement algorithm. The proposed algorithm consists of the following two steps:

1. Filling the holes in the estimated disparity image, and
2. Sharpening the edges and object boundaries.

Since we obtained the disparity image based on down-sampled reference images, we bring it back to the original size while performing refinement. The estimated disparity image from the previous steps has some holes due to the occlusion and mismatches which need to be filled. The hole regions usually belong to the background which cannot be seen from the other reference view. Hence, the algorithm fails to estimate a depth value for those specific regions. To fill up the holes, we first select the pixels that belong to the background among the neighboring non-zero pixels of the hole region. The holes are then filled by a weighted average on the selected correct pixels using Eq. (12).

$$d_{\text{hole}} = \frac{\sum_{i=1}^{N} w_i d_i^{\text{bg}}}{\sum_{i=1}^{N} w_i} \tag{12}$$

where d_i^{bg} is the background depth value and w_i is the weighting factor based on the distance from the background depth pixel to the current hole.

The weights are calculated using Gaussian distribution based on the distance to the current pixel using Eq. (14). Therefore, the farther pixels would have less impact on the calculated depth value. The output of the hole filling algorithm is a low-resolution dense disparity image which needs to be up-sampled to the original size. The up-sampling is performed by applying a trilateral filter which makes the boundaries sharper and corrects the misaligned regions using Eq. (13). The designed filter consists of three terms: depth data, texture data and the curvature.

$$d\,(p) = \frac{1}{w} \sum d\,(q)\,.f_{\text{dep}}\left(\|\, d_p - d_q \,\|\right).f_{\text{tex}}\left(\|\, C_p - C_q \,\|\right).f_{\text{curv}}\left(\|\, k_p - k_q \,\|\right) \tag{13}$$

where $\|\,.\,\|$ is the Euclidean distance between two pixels and d is the disparity value. C and k are the color and curvature values, respectively. f is the Gaussian distribution with standard deviation σ defined by Eq. (14).

$$f(x) = e^{-(\|x\|/\sigma)^2} \tag{14}$$

We also ensure that no new depth values are introduced during the up-sampling. Therefore, when the disparity image is filtered using the trilateral filter, the new depth values are adjusted by mapping them to the nearest depth value which already exists in the disparity image.

4 Experimental results

In this section, we present simulation results to show the efficiency of the proposed method compared with those of the state-of-the-art SGM-based stereo matching methods.

To evaluate the performance of our depth estimation and refinement algorithms, we used the Middlebury [36], KITTI stereovision benchmark suite [37], and the Daimler dataset [38]. The workstation runs the Windows 7 operating system with Intel Xeon Quad-Core processor and 8 GB RAM.

To evaluate the performance of the proposed stereo matching method, we compute the error statistics with respect to the ground truth depth map. The percentage of bad matching pixels is a common quality measure used to compare the performance of different depth estimation methods.

The performance of the proposed method is compared with the following reference systems. The method proposed in [20] is the original SGM which uses pixelwise matching of mutual information and approximates a global smoothness constraint by combining multiple 1D constraints. The stereo matching proposed in [39] uses a sparse Census mask and the algorithm is implemented on embedded systems. PlaneFitSGM [40] is based on a Census-based cost calculation and performs the disparity optimization on horizontal stripes of the image. Gradient-based Census [41] performs correlation-based stereo matching and is suitable for the deployment in embedded real-time systems.

Four different image pairs from the Middlebury dataset [36] are chosen for the evaluation. Figure 9 compares the quality of depth maps generated by the proposed method and the references. The use of multiple Census window patterns and efficient depth refinement technique by considering the edges of objects makes the proposed method comparable to other methods of SGM family.

Table 1 indicates the error statistics of percentage of bad pixels with respect to the provided ground truth depth map by Middlebury benchmark [36] for different algorithm configurations. The percentage of bad pixels evaluation criterion is defined by Eq. (15).

$$P_{bp} = \frac{1}{N} \sum \left(\left| d_{\mathrm{G}}(x, y) - d_{\mathrm{GT}}(x, y) \right| > \sigma_d \right), \tag{15}$$

where d_{G} and d_{GT} are the generated and ground truth depth values, respectively. σ is the error tolerance and N is the total number of pixels of the image.

The percentage of correctly matched pixels for different Census-based matching techniques is shown in Fig. 10. Avoiding any new depth value in the refinement step

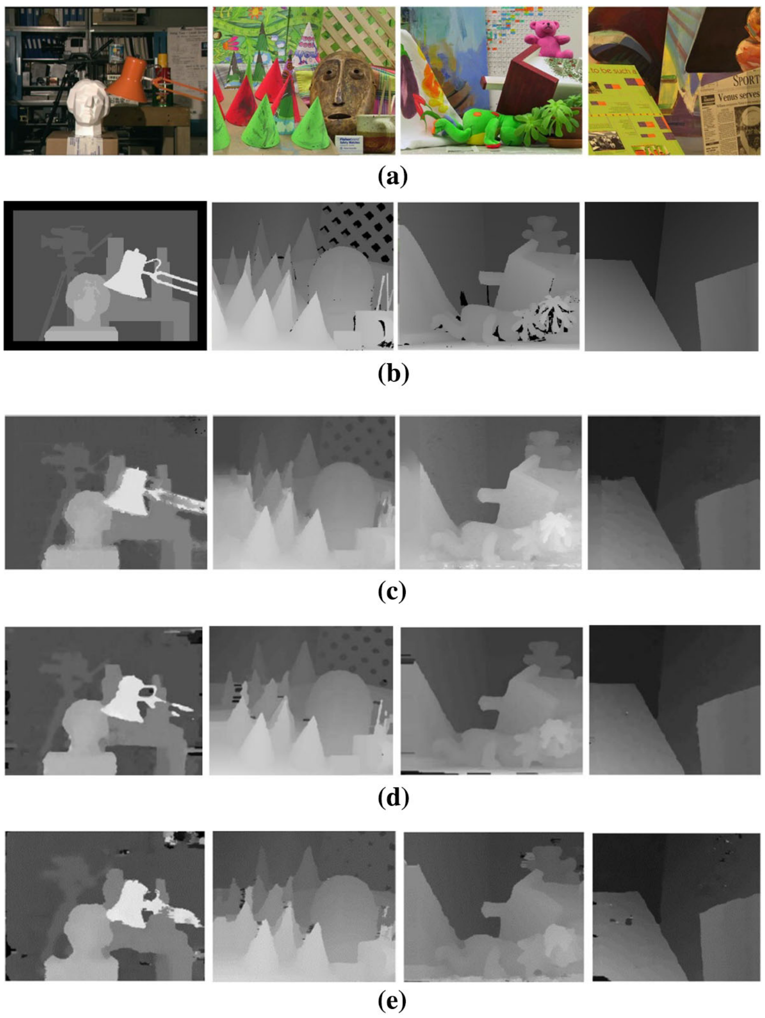


Fig. 9 Visual comparison of depth maps generated by different algorithms using Middlebury dataset [36]: **a** the original test images, **b** ground truth depth maps, and depth maps obtained by **c** proposed method, **d** PlaneFitSGM [40], **e** gradient-based Census [41]

of the proposed method significantly reduces the percentage of bad pixel compared with the methods in [20,39,40].

In addition to this, applying Census transform as a matching metric helps us to find the perfect match for each pixel and increases the percentage of correctly matched pixels.

Table 2 compares the processing time of the proposed algorithm with other state-of-the-art techniques. The experiments are done by running the algorithms on the Middlebury dataset [36] using C programming on CPU.

Applying adaptive Census window patterns reduces the overall complexity of the algorithm and leads to achieving higher frame rates.

Table 1 Percentage of bad pixels for the Middlebury dataset [36]

Stereo image	Proposed method (%)	SGM [20] (%)	Census [39] (%)	PlaneFitSGM [40] (%)
Tsukuba	**2.42**	3.96	6.25	4.2
Cones	**7.74**	9.75	9.54	9.26
Cones	**11.77**	12.2	13.8	11.6
Venus	**0.85**	1.57	2.42	1.87

The result of the proposed method is written in bold for better visualization

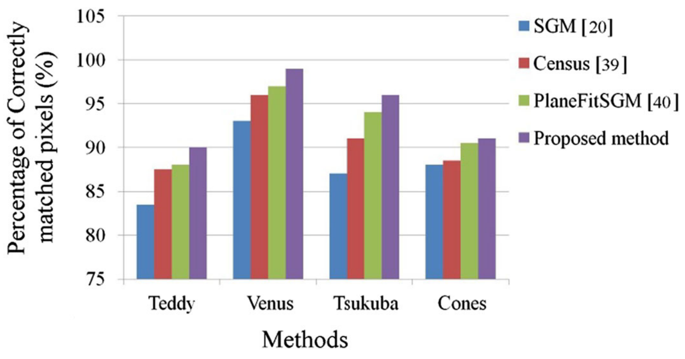


Fig. 10 Percentage of correctly matched pixels: comparison between the proposed method and algorithms in [20,39,40]

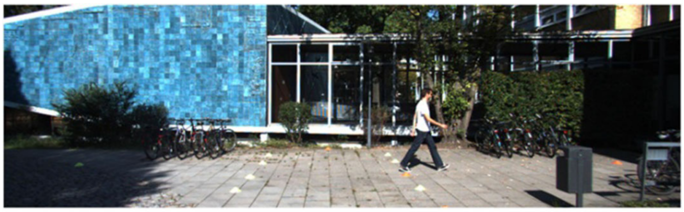
Table 2 Computational time complexity

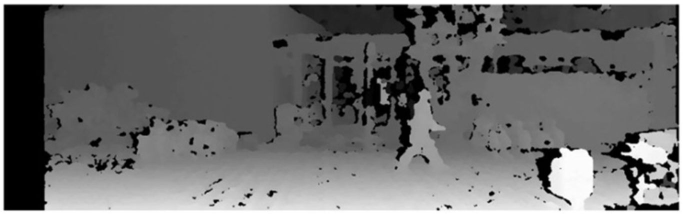
Stereo image	Tsukuba	Cones	Teddy	Venus
Size (pixels)	384 × 288	450 × 375	450 × 375	434 × 383
Proposed method (ms)	**50**	**57**	**61**	**51**
SGM [20] (ms)	64	73	73	68
Census [39] (ms)	83	91	94	87
PlanFitSGM [40] (ms)	3561	6351	9561	3824

The result of the proposed method is written in bold for better visualization

The proposed algorithm has been tested on KITTI dataset [37] which consists of 194 training image pairs and 195 test image pairs. The images have 1224 × 370 pixels resolution. Figure 11 shows the result of depth estimation and refinement for a sample left side image of KITTI dataset [37]. The holes are filled in the refined depth map and objects have much sharper boundaries.

As stated earlier, the proposed depth refinement algorithm uses neighboring background pixels solely to fill the holes and also incorporates a depth adjustment stage to ensure that no new depth values are replaced a correct depth value in the depth map. Figure 12 compares the result of hole filling where in Fig. 12b only the background depth pixels are used. Considering only the background pixels for depth refinement avoids several artifacts such as shrinking of depth values around object boundaries (similar to the case around the pedestrian and the trash can in Fig. 12b). One should


(a)


(b)

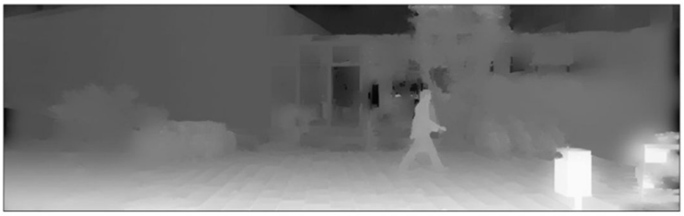
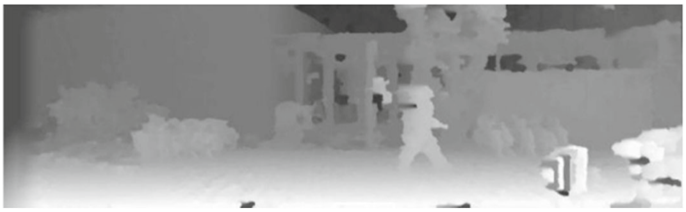

(c)

Fig. 11 Proposed method output: **a** color image and **b** estimated depth map, and **c** refined depth map using KITTI dataset [37] (color figure online)


(a)

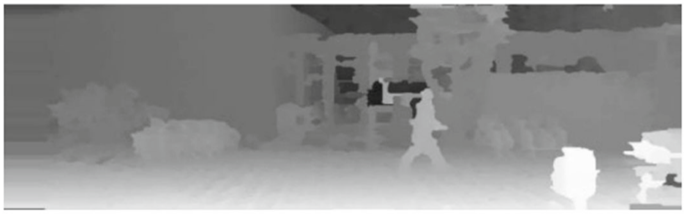

(b)

Fig. 12 Hole filling comparison using KITTI dataset [37]: **a** filling by all the surrounding correct pixels, **b** only considering background depth pixels

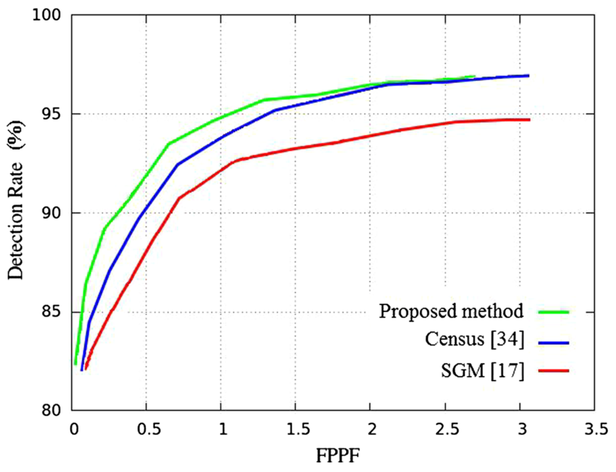


Fig. 13 ROC curve comparison between different algorithms using Diamler dataset [38]

keep in mind that Fig. 12 only shows the hole filling result without applying edge sharpening.

For a more complex scenario where several number of pedestrians is present in the scene, it becomes more challenging to derive a clear depth map for each pedestrian individually. Figure 14 shows the result of the proposed depth estimation method on four different images of KITTI dataset [37] under this challenging situation.

Since the main target application of the proposed method is pedestrian detection systems, we provide additional results based on pedestrian detection rate. We apply our proposed depth estimation method to the stereo-based pedestrian detection algorithm presented in [2]. The proposed method in [2] uses an adaptive window for region of interest (ROI) generation using depth maps. Also, Support Vector Machine (SVM) is used to classify the ROIs into pedestrian and non-pedestrian classes. The Daimler dataset [38] has been used for this part of experimental results. It consists of 21790 image pairs with size of 640×480 pixels captured from a stereo vision camera mounted on a vehicle.

To calculate the ROC curve, we plot the detection rate versus the average of false-positive per frame (FPPF) using Eq. 16.

$$\begin{aligned}\text{Detection rate} &= \frac{\text{TP}}{\text{TP} + \text{FN}},\\ \text{FPPF} &= \frac{\text{TFPP}}{\#\,\text{frames}},\end{aligned} \tag{16}$$

where TP, FN and FP are true positive, false negative and false positive, respectively.

The calculated ROC curves are shown in Fig. 13. The detection rates are improved using the proposed method. In other words, for a fixed false-positive probability, we get higher detection rate.

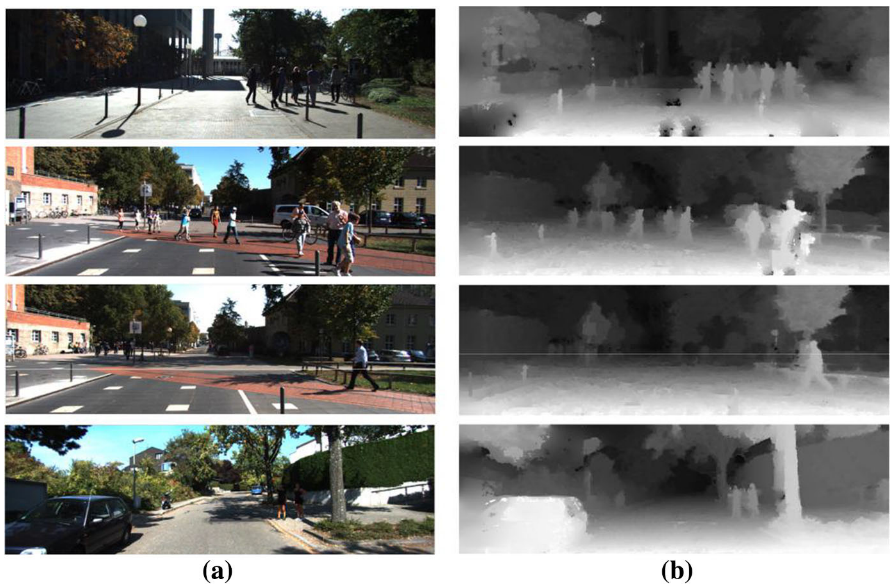


Fig. 14 Depth estimation by the proposed method on real-world stereo images. **a** *Left* color images. **b** Estimated depth map (color figure online)

5 Conclusion

In this paper, a novel depth estimation algorithm has been proposed. The proposed method is based on adaptive window patterns of Census transform which make it robust against illumination changes and suitable for applications like advanced driver assistance systems. By down sampling the reference images, the computational complexity of the whole algorithm is reduced. A modified cross-based cost aggregation technique is proposed that generated cross-shape support regions for each pixel individually. A stripe-based optimization path is used in finding the best match by avoiding the error propagation into unwanted regions. The proposed depth refinement technique aims at filling the holes and sharpening the object boundaries. The background depth pixels are used to fill the holes of the estimated depth map and the proposed trilateral filter is used to enhance the quality of the depth map. Simulation results indicate that the proposed methods fulfill the aims by improving the quality of the generated depth maps and reducing the computational complexity (Fig. 14).

References

1. Gavrila DM, Munder S (2007) Multi-cue pedestrian detection and tracking from a moving vehicle. Int J Comput Vis 73:41–59
2. Mesmakhosroshahi M, Kim J, Lee Y, Kim J-B (2013) Stereo based region of interest generation for pedestrian detection in driver assistance systems. In: Proceedings of the International Conference on Image Processing (ICIP), pp 3386–3389
3. Kim J, Mesmakhosroshahi M (2015) Stereo-based region of interest generation for real-time pedestrian detection. Peer-to Peer Netw Appl 8:181–188

4. Hirschmuller H, Scharstein D (2007) Evaluation of cost functions for stereo matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Minneapolis, 18–23 June 2007, pp 1–8
5. Zhang Z (2012) Microsoft Kinect sensor and its effect. IEEE Multim 19(2):4–10
6. Boykov Y, Veksler O, Zabih R (1998) A variable window approach to early vision. IEEE Trans Pattern Anal Mach Intell 20(12):1283–1294
7. Kang SB, Szeliski R, Jinxjang C (2001) Handling occlusions in dense multi-view stereo. In: Proceedings of 2001 IEEE Society Conference on Computer Vision and Pattern Recognition (CVPR), vol 1. pp 103–110
8. Sun J, Zheng NN, Shum HY (2003) Stereo matching using belief propagation. IEEE Trans Pattern Anal Mach Intell 25(7):399–406
9. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization with graph cuts. IEEE Trans Pattern Anal Mach Intell 23(11):1222–1239
10. Gong M, Yang Y (2003) Fast stereo matching using reliability-based dynamic programming and consistency constrains. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer, pp 610–617
11. Hirschmuller H (2005) Accurate and efficient stereo processing by semi-global matching and mutual information. In: CVPR '05 Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol 2. IEEE Computer Society, Washington, pp 807–814
12. Vijayanagar KR, Loghman M, Kim J (2012) Refinement of depth maps generated by low-cost depth sensors. In: Proceedings of International SoC Design Conference (ISOCC), pp 533–536
13. Vijayanagar KR, Loghman M, Kim J (2014) Real-time refinement of kinect depth maps using multi-resolution anisotropic diffusion. Mob Netw Appl 19(3):414–425
14. Scharstein D, Szeleiski R (2002) A taxonomy and evaluation of dense two frame stereo correspondence algorithm. Int J Comput Vis 47(1):7–42
15. Yang R, Pollefeys M (2003) Multi-resolution real-time stereo on commodity graphics hardware. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp 211–218
16. Loghman M, Zarshenas A, Chung K-H, Lee Y, Kim J (2014) A novel depth estimation method for uncalibrated stereo images. In: Proceedings of International SoC Design Conference (ISOCC), pp 186–187
17. Yang R, Pollefeys M (2004) Improved real-time stereo on commodity graphics hard-ware. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop on Real-time 3D Sensors and Their Use
18. Gong M, Yang Y-H (2005) Near real-time reliable stereo matching using programmable graphics hardware. In: CVPR '05 Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol 1. IEEE Computer Society, Washington, pp 924–931
19. Gong M, Yang R (2005) Image-gradient-guided real-time stereo on graphics hardware. In: Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM'05), IEEE, pp 548–555
20. Ernst I, Hirschmuller H (2008) Mutual information based semi-global stereo matching on the GPU. In: Bebis G, Boyle R et al (eds) Advances in visual computing, vol 5358. Lecture notes in computer science, Springer, Berlin, pp 228–239
21. Gehrig SK, Eberli F, Meyer T (2009) A real-time low-power stereo vision engine using semi-global matching. In: Mario Fritz M, Schiele B, Piater JH (eds) Computer Vision Systems, 7th International Conference on Computer Vision Systems, ICVS 2009, 13–15 Oct 2009 Liège, Belgium, pp 134–143
22. Hermann S, Klette R, Destefanis E (2009) Inclusion of a second-order prior into semi-global matching. In: The 3rd Pacific-Rim Symposium on Image and Video Technology (PSIVT2009) Tokyo, 13–16 Jan 2009
23. Hermann S, Morales S, Klett R (2011) Half-resolution semi-global stereo matching. In: Proceedings of IEEE Intelligent Vehicles Symposium, pp 201–206
24. Haller I, Pantillie C, Oniga F, Nedevschi S (2010) Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In: Proceedings of IEEE Intelligent Vehicles Symposium, pp 369–376
25. Hermann S, Klette R (2013) Iterative semi-global matching for robust deriver assistance systems. Lect Notes Comput Sci 7726:465–478
26. Banz C, Blume H, Pirsch P (2011) Real-time semi-global matching disparity estimation on GPU. In: IEEE International Conference on Computer Vision Workshop, pp 514–521

27. Banz C, Hesselbarth S, Flatt H, Blume H (2010) Real-time stereo vision system using semi-global matching disparity estimation: architecture and FPGA-implementation. In: Proceedings of International Conference on Embedded Computer Systems, pp 93–101
28. Gehrig SK, Rabe C (2010) Real-time semi-global matching on the GPU. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition Workshop, pp 85–92
29. Pantilie CD, Nedevschi S (2011) Real-time semi-global matching using segmentation and plane fitting for improved accuracy on the GPU. In: 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 5–7 Oct 2011 pp 784–790
30. Pantilie CD, Nedevschi S (2012) SORT-SGM: Subpixel optimized real-time semi-global matching for intelligent vehicles. IEEE Trans Veh Techn 61(3):1032–1042
31. Loghman M, Chung K-H, Lee Y, Kim J (2014) Depth map estimation using modified Census transform and semi-global matching. In: Proceedings of International SoC Design Conference (ISOCC), pp 158–159
32. Loghman M, Kim J (2013) SGM-based dense disparity estimation using adaptive census transform. In: Proceedings of IEEE International Conference on Connected Vehicles and Expo (ICCVE), pp 592–597
33. Tombari F, Mattoccia S, Stefano LD (2006) Segmentation-based adaptive support for accurate stereo correspondence. Advances in image and video technology. Vol. 4872. Springer, New York, pp 427–438
34. Hosni A, Bleyer M, Gelautz M, Rheman C (2009) Local stereo matching using geodesic support weights. In: IEEE International Conference on Image Processing (ICIP), 2093–2096
35. Zhang K, Lu J, Lafruit G (2009) Cross-based local stereo matching using orthogonal integral images. IEEE Trans Circuits Syst Video Technol 19(7):1073–1079
36. Scharstein D, Szeliske R (2003) High-accuracy stereo depth maps using structured light. In: CVPR'03 Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, pp 195–202
37. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR), Providence, USA
38. Keller CG, Enzweiler M, Gavrila DM (2011) A new benchmark for stereo-based pedestrian detection. In: Proceedings of the Intelligent Vehicles Symposium, pp 691–696
39. Humenberger M, Zinner C, Weber M, Kubinger W, Vincze M (2010) A fast stereo matching algorithm suitable for embedded real-time systems. Comput Vis Image Underst 114(11):1180–1202
40. Humenberger M, Engelke T, Kubinger W (2010) A census-based stereo vision algorithm using modified semi-global matching and plane-fitting to improve matching quality. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 77–84
41. Ambrosch K, Kubinger W (2010) Accurate hardware-based stereo vision. Comput Vis Image Underst 114(11):1303–1316