

# Fast Stereo Matching Fully Utilizing Super-Pixels

Masayuki Miyama

Division of Electrical Engineering and Computer Science, Kanazawa University, Kanazawa, Ishikawa, Japan

Email: miyama@se.kanazawa-u.ac.jp

**How to cite this paper:** Miyama, M. (2018) Fast Stereo Matching Fully Utilizing Super-Pixels. *Journal of Computer and Communications*, 6, 15-27.  
<https://doi.org/10.4236/jcc.2018.68002>

**Received:** July 9, 2018

**Accepted:** July 31, 2018

**Published:** August 2, 2018

Copyright © 2018 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

In this paper, we propose a depth image generation method by stereo matching on super-pixel (SP) basis. In the proposed method, block matching is performed only at the center of the SP, and the obtained disparity is applied to all pixels of the SP. Next, in order to improve the disparity, a new SP-based cost filter is introduced. This filter multiplies the matching cost of the surrounding SP by a weight based on reliability and similarity and sums the weighted costs of neighbors. In addition, we propose two new error checking methods. One-way check uses only a unidirectional disparity estimation with a small amount of calculation to detect errors. Cross recovery uses cross checking and error recovery to repair lacks of objects that are problematic with SP-based matching. As a result of the experiment, the execution time of the proposed method using the one-way check was about 1/100 of the full search, and the accuracy was almost equivalent. The accuracy using cross recovery exceeded the full search, and the execution time was about 1/60. Speeding up while maintaining accuracy increases the application range of depth images.

## Keywords

Stereo Matching, Super-Pixel, Cost Filter, Cross Check, One-Way Check

## 1. Introduction

The depth image is an image in which the distance to the object in the three-dimensional space is projected as shading or color on the imaging plane set at the observer's viewpoint. Depth images are used for driving safety support, automatic driving, autonomous robot, user interface, monitoring and so on. Distance measurement using millimeter waves or laser light is good in accuracy, but it depends on physical properties of object and is susceptible to disturbance.

On the other hand, in the method using a stereo camera, disparity is obtained from two images captured by two cameras on the left and right. Then, based on the principle of triangulation, we obtain the distance from the camera to the object using the disparity. In the stereo method, matching between right and left pixels is necessary, and the amount of calculation is huge. For this reason, compatibility between accuracy and processing speed is a problem of stereo matching.

So we introduce super-pixels (SPs) for the depth image generation. An SP is a small area formed by collecting adjacent similar pixels. SP segmentation over-divides the inside of the object, but extracts object boundaries well. Processing on SP basis has an overwhelmingly small amount of calculation as compared with processing on pixel basis. By using SPs for depth image generation, depth images with clear object boundaries can be created at high speed.

This paper newly proposes a fast method of depth image generation using SPs. To reduce the amount of computation, we use an existing method in which block matching is performed only at the center of the SP and the obtained disparity is applied to all pixels in the SP [1]. Then, filtering of matching cost on SP basis is newly proposed to improve disparities. Furthermore, in contrast to usual cross check which requires disparity estimations in both left and right directions, we newly introduce one-way check that detects errors only using unidirectional disparity estimation. We also propose a new method called cross recovery that restores lacks of objects problematic in SP-based matching. As a result of the experiment, the execution time of the proposed method with the one-way check was about 1/160 of full search, and the accuracy was equivalent. The accuracy using the cross recovery exceeded the full search, and the execution time was about 1/60. Compared with the state of the art using pixel-based cost filtering, the speed becomes 15 times or more, though the accuracy decreases a little.

In contrast to the conventional methods on pixel basis, this work fully utilizes SPs to speed up disparity estimation while maintaining accuracy. For this purpose, we newly propose SP-based cost filter, one-way check, and cross recovery, which are contributions of this work. In depth image generation, speedup while keeping accuracy is important for real time processing, resolution enhancement, and search range expansion. This will extend the application scope of the depth image. In this paper, we added the cross recovery proposal and a comparison with the state of the art to our previous work [2].

The structure of this paper is as follows. Section 2 describes related work. Section 3 details the proposed method. Section 4 explains the experiment setup and results. Section 5 concludes this paper.

## 2. Related Work

Many researches on depth image generation by stereo matching have been conducted. A survey of these researches has been performed [3]. This survey also provides the benchmark data and evaluation programs used for comparing var-

ious algorithms, and they are released in public.

The method of stereo matching is roughly divided into local method and global method. The local method estimates disparity using only pixels around the target pixel. Block matching, which is the most basic method, sets a support region in a block shape around the pixel of interest. A block image most similar to the support block image is searched from the opposite image, and the distance from the center of the similar block to the original position is taken as a disparity.

At this time, if the support area contains another object different from the object to which the target pixel belongs, it will easily cause an erroneous estimation. A method of collecting only similar pixels surrounding the target pixel in a cross shape to construct the support area, and efficiently aggregating matching costs between pixels in the area has been proposed [4]. A method, multiplying the matching cost of the surrounding pixel by the weight corresponding to the similarity and the distance to the target pixel, aggregating them in the neighbor, and finding the disparity having the minimum cost, has been proposed [5] [6]. These are high precision and high speed, but they are pixel basis processing. The support area is set to every pixel and not usually used for other purposes.

The local methods often adopt simple error detection by cross checking. The cross checking is used to detect errors such as occlusion, and finds the difference between disparities of the corresponding left and right pixels. Error pixels are filled with the smaller disparity on either the left or the right.

The global method performs disparity estimation with global optimization of the entire image. Dynamic programming is typical for the optimization. In this method, disparities are optimized under the constraint that the arrangement order of the corresponding pixels of the left and right images does not intersect. A method to increase the speed by limiting the constraint to 8 directions has been proposed [7]. Besides this, a method based on graph cut [8] and a method using belief propagation [9] are known.

The global methods often adopt error detection with global optimization. After optimization by belief propagation, an error detection method using constraints by warping and cross check has been proposed [10]. After global stereo matching, an error detection method using ordering, uniqueness, and color similarity constraints has been proposed [11]. A method of alternately performing disparity estimation and occlusion detection by belief propagation using visibility constraint has been proposed [12]. Compared to the local methods, global methods are disadvantageous for speeding up.

Methods with SPs for depth image generation have been proposed. For reducing computational complexity, a way to perform block matching only on the center of the SP rather than the whole image has been proposed [1]. The use of SPs in this method is only pixel sampling to estimate disparity. They don't use the SPs either to create support areas or to improve disparities.

A method for improving the disparity model of an SP with the Markov ran-

dom field consisting of the connection of adjacent SPs has been proposed [13]. This method uses the semi-global method to obtain the pixel disparities, and calculates an initial model with the all disparities in an SP. A method, in which the disparities of all the pixels in the SP is obtained by the semi-global method, the median value is taken as the initial value of the SP disparity, and the SP disparity is improved by filtering using the surrounding SP disparities, has been proposed [14]. In order to clarify the object boundary, a method of adding filtering in SP basis after the conventional pixel-wise cost filtering has been proposed [15], but the execution time increases. An SP-based method explicitly using graph-based energy minimization to deal with occlusion has been proposed, but it is slow as it is a global method [16]. These methods use SPs only for disparity improvement. In contrast, the proposed method fully utilizes SPs for all of pixel sampling, support area creation, and disparity improvement.

### 3. Proposed Method

This section explains the details of the proposed method. The procedure of the method with the one-way check is as follows.

- 1) SP segmentation;
- 2) Center search;
- 3) One-way check;
- 4) Reliability calculation;
- 5) Smoothing with cost filter;
- 6) One-way check on SP basis;
- 7) Gap filling.

The procedure of the method with the cross check and recovery is as follows.

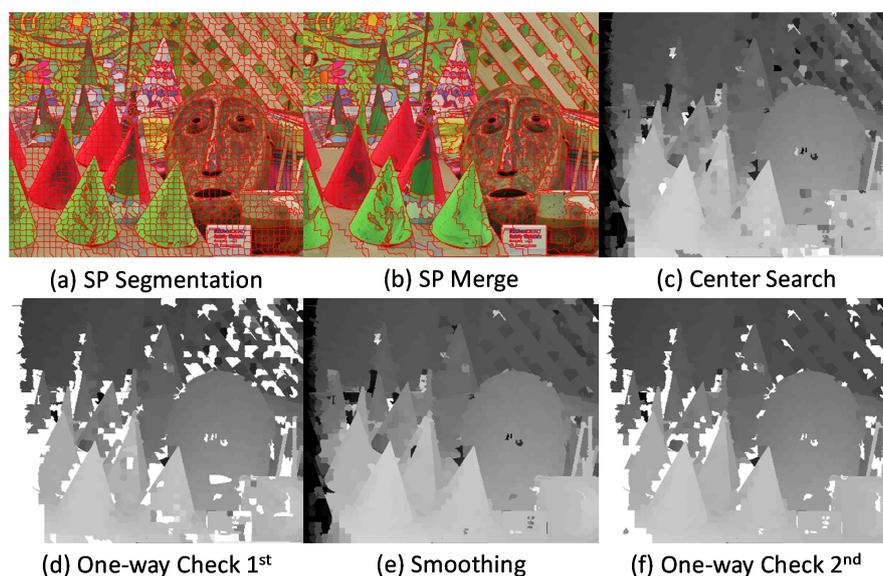
- 1) SP segmentation;
- 2) Center search;
- 3) Cross check;
- 4) Reliability calculation;
- 5) Smoothing with cost filter;
- 6) Cross check and recovery;
- 7) Gap filling.

A process of a test image cone is illustrated in **Figure 1**. The SP segmentation divides the image into SPs. For the purpose, we adopt Simplified SEEDS (SS) [17]. SS initially divides the image into a lattice shape and repeats boundary update using color similarity. SS has high speed and high segmentation accuracy. A segmentation result of the cone by SS is shown in **Figure 1(a)**.

For the center search, block matching is performed only at the center of the SP [1]. The disparity of the center pixel is used as the disparity of the entire super-pixel. The formula for the center search is as follows.

$$raw(R, d) = \sum_{(u,v) \in S_R} \sum_{c \in C} |I_c(x_R + u, y_R + v) - J_c(x_R + u + d, y_R + v)| \quad (1)$$

Here,  $d$  is a disparity,  $S_R$  is a support area of  $R$ ,  $u, v$  are horizontal and vertical coordinates in  $S_R$ ,  $x_R, y_R$  are the center coordinates of  $R$ ,  $I$  represents a



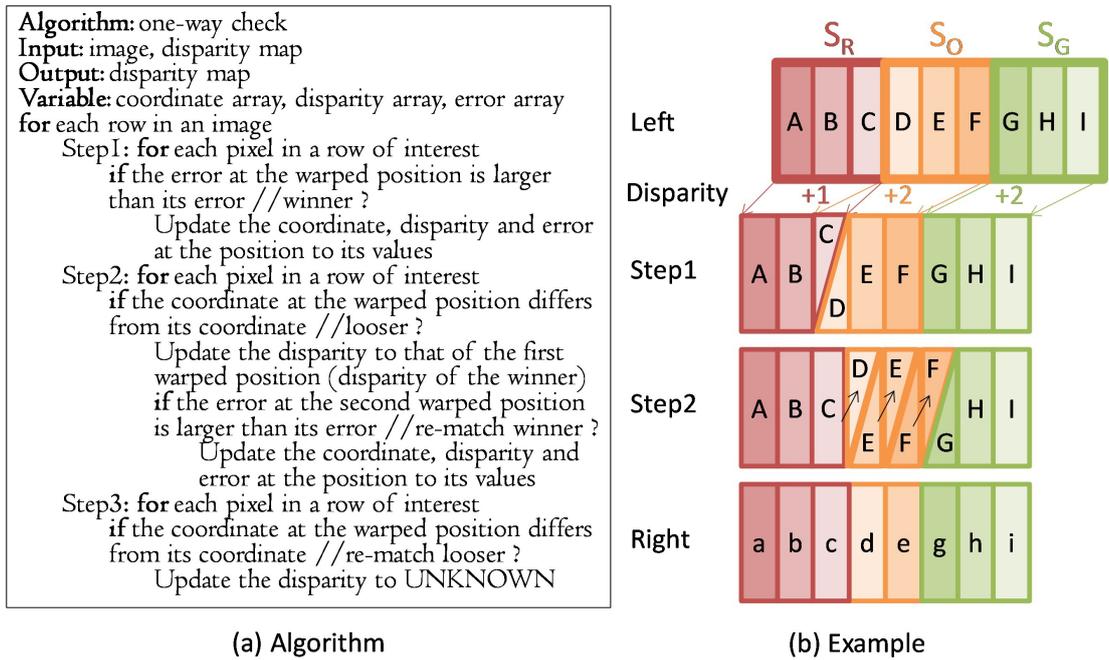
**Figure 1.** Proposed algorithm.

target image (left image), and  $J$  represents a reference image (right image). As expressed by this formula, the sum of absolute difference is used as the matching cost. In **Figure 1(c)**, there are variations in disparities for each object due to erroneous estimation caused by noise, and the object images are mottled.

In the one-way check, results of only unidirectional matching are used to detect the erroneous estimations of disparities. The one-way check algorithm is shown in **Figure 2(a)** and explained using the example of **Figure 2(b)**. Suppose three SPs of red, orange, and green are lined up in the left image. As a result of the center search, the red disparity is 1, the orange one is 2, and the green one is 2. Notice that pixel F is in occlusion, and the orange disparity is erroneously estimated. In step 1, multiple left pixels may match the same right pixel. In this example, the left pixels C and D match the same right pixel C. Then, the error between the right pixel and the left pixel is calculated, and the pixel with the smallest error is taken as the winner. In this example, since the color similarity between the right pixel C and the left pixel C is high and the error is small, the left pixel C is the winner for the right pixel C. In step 2, the loser D makes a re-match with the right pixel D pointed by the disparity (1) of the winner C. As a result, since the error is smaller than that of the left pixel E, the left pixel D is the winner for the right pixel D. Repetition of re-matches reveals that all pixels in the orange area were false estimates.

As can be seen from this example, the SP of the background, partially hidden by the foreground SP, is often presumed to be the same disparity of the foreground SP. At this time, it is impossible to detect all erroneous detections with step 1 alone. In order to find erroneous disparity estimations of all the pixels in the background SP, re-matching the loser warped with the disparity of the winner in step 2 is necessary.

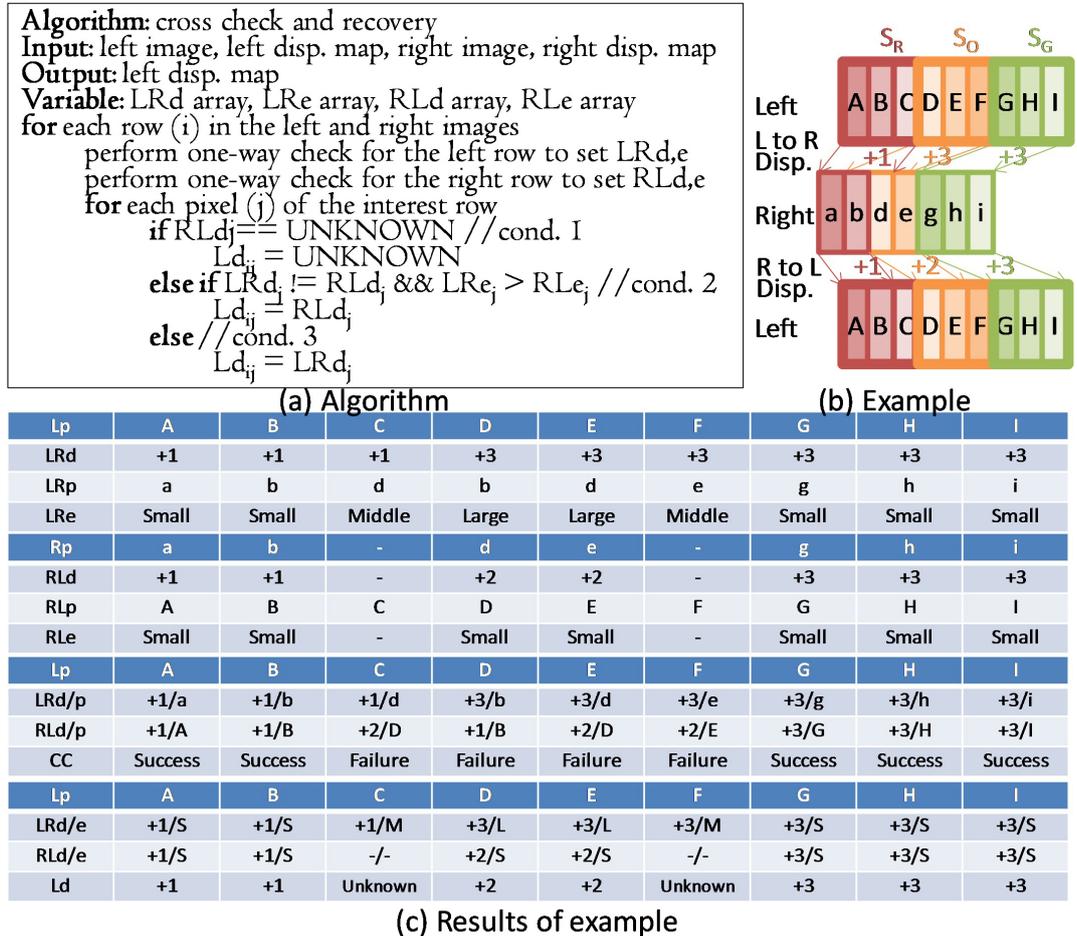
In this check, the pixel-wise error is used for the correspondence judgment of



**Figure 2.** Onw-way check algorithm and its example.

the left and right pixels, and the support area is not used. For this reason, the judgment is easy to make a mistake. Therefore, aggregation is performed on SP basis, and if the ratio of the total number of erroneously estimated pixels to all pixels is larger than the threshold  $TH_{check}$  erroneous estimation of the SP is determined. As shown in **Figure 1(d)**, erroneous estimations due to occlusion and texture shortage are detected as a result of the one-way check.

In the estimation based on SP, lacks often occur in part of objects due to erroneous estimation. The cross recovery is the way to repair the lacks. The algorithm of the cross recovery is shown in **Figure 3(a)** and explained using the example of **Figure 3(b)**. The disparity of the R region is +1, O region is +2, and G region is +3. Notice that pixel C and F are occluded. As a result of the center search from left to right, it is estimated that the disparity of R region is +1, O region is +3, and G region is +3. In the case of right to left, the disparity of R region is +1, O region is +2, and G region is +3. The top table of **Figure 3(c)** shows the correspondence of left to right disparity (LRd), right pixel (LRp), and error (LRe) with respect to the left pixel (Lp). The second table shows the correspondence of right to left disparity (RLd), left pixel (RLp), and error (RLe) with respect to the right pixel (Rp). The third table shows the result of the usual cross check (CC). For example, although the disparity of C corresponding to D is +1, since the disparity of d corresponding to D is +2, the check fails. Since the pixels C to F become undefined due to the check failure, the disparity of O region is incorrectly estimated as +1 after the filling of disparities. The bottom table is the result of the cross recovery. Consider pixel C first. There is no right pixel to be warped to the left pixel C, and the RLd of C is unknown. When the RLd is unknown (condition 1), the disparity Ld is determined to be unknown.



**Figure 3.** Cross recovery algorithm and its example.

Consider pixel D next. Since the warp destination of the pixel D is the pixel b, the error LRe is large. If the LRd and the RLd are different (condition 2), then the disparity with the smaller error is adopted, the disparity Ld of D becomes +2 (RLd in this case). Finally, when neither condition 1 nor condition 2 holds (LRd and RLd are the same), LRd is adopted. As a result of the cross recovery, the disparity of O region is estimated correctly as +2.

In the smoothing, the matching cost is corrected by multiplying the cost of the surrounding SP and the weight consisted of similarity and reliability, and summing them up in the neighbor. The cost filter is expressed by the following equations.

$$sim(R, Q) = \exp\left(-\alpha \sqrt{\sum_{c \in C} (ave_c(R) - ave_c(Q))^2}\right) \quad (2)$$

$$rel(R) = \frac{1}{N_R} \sum_{i \in R} is\_correct(i) \quad (3)$$

$$wei(R, Q) = sim(R, Q) \times rel(Q) \quad (4)$$

$$fil(R, d) = \frac{\sum_{Q \in N_R} wei(R, Q) \times raw(Q, d)}{\sum_{Q \in N_R} wei(R, Q)} \quad (5)$$

$$dis(R) = \arg \min_{d \in D} fil(R, d) \quad (6)$$

Here,  $sim(R, Q)$  is the similarity by the averaged color of  $R$  and  $Q$ ,  $R$  is the SP of interest,  $Q$  is the surrounding SP,  $\alpha$  is a similarity coefficient,  $c$  is a color component, and  $C$  is a set of color components.  $N_R$  is the number of pixels in  $R$ ,  $is\_correct(i)$  is a function that returns 0 if the one-way check on pixel  $i$  is erroneously estimated, and returns 1 otherwise.  $rel(R)$  is the reliability of  $R$  depending on the ratio of the number of correct pixels determined by the one-way check to the total number of pixels in  $R$ .  $wei(R, Q)$  is the weight between the surrounding SP  $Q$  and the target SP  $R$ . In the cost filter, the center search cost for the disparity  $d$  of the SP  $Q$  is multiplied by the weight  $wei(R, Q)$  based on the reliability and the similarity. Then we sum up them for all SPs surrounding  $R$  expressed by  $N_R$ .  $fil(R, d)$  is the cost after the filtering. A disparity  $d$  of  $R$  which minimizes  $fil(R, d)$  in the range  $D$  is  $dis(R)$ . As a result of the filtering, the mottling pattern is almost gone in **Figure 1(e)**. This SP-based filter is similar to the conventional cost filter [5] [6], but that is pixel-wise processing.

Then we perform the error check again. The erroneous estimation is reduced more than the first check in **Figure 1(f)**. Finally, the pixel erroneously estimated is filled with the smaller disparity among the correct right and left pixels closest to the pixel, as shown in **Figure 4(e)**.

## 4. Experiments

### 4.1. Setup

Four images (Cones, Teddy, Tsukuba, Venus) of Middlebury Stereo Dataset for the test image and StereoMatcher for the evaluation software were used [3]. RMS (Root Mean Square) and Bad Pixel (proportion of pixels having an error larger than 1 against to the whole of pixels) were adopted as the evaluation metrics. The algorithm to compare is as follows.

- FNC15: Full search (T:merge, S:merge,  $15 \times 15$ ) + cross check;
- CNC07: Center search (T:SP, S:block,  $7 \times 7$ ) + cross check;
- CSC07: Center search (T:SP, S:block,  $7 \times 7$ ) + smoothing + cross check;
- CSR07: Center search (T:SP, S:block,  $7 \times 7$ ) + smoothing + cross recovery;
- CSO07: Center search (T:SP, S:block,  $7 \times 7$ ) + smoothing + one-way check;
- CVF: Cost volume filter.

In FNC15, overlap between  $15 \times 15$  block and an area created by merging SPs (see **Figure 1(b)**) was taken as a support area of the template (T). We also created the support area of the search window (S) in the same way. In the four methods using the center search, the overlap of the  $7 \times 7$  block and SP is set as the support area of the template, and the whole  $7 \times 7$  block is set as the support area of the search window. The initial lattice partition size of the SP was  $7 \times 7$ , the similarity coefficient  $\alpha$  of the weight calculation was 0.2, and the one-way check on SP basis threshold  $TH_{check}$  was 0.6. Experiments of these five methods used our own program. For the experiment of CVF, we used the program

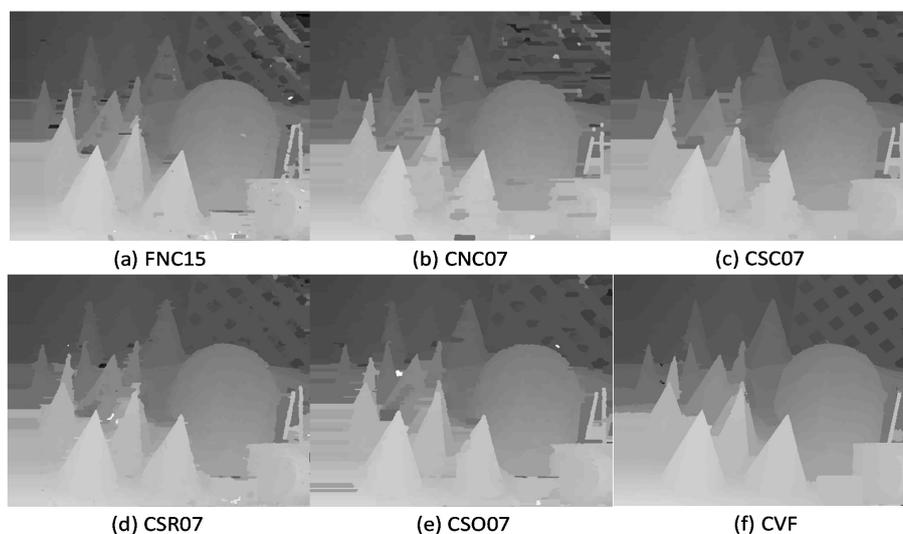
created by the authors of the paper [6]. We used Microsoft VisualStudio 2013 for the software development. The OS of the PC that executed the program was Windows 10, the CPU was Intel Core i5, the operating frequency of the CPU was 3 GHz, and the memory capacity was 4 GB.

## 4.2. Results

The result of the cones is shown in **Figure 4**. Because the merged segmentation map is used to create the support area, only the pixels in the same object are matched, so the object boundary of FNC15 is clear. With the small block size  $7 \times 7$ , disparities in the same object vary due to matching failures, and a mottled pattern is observed in CNC07. CSC07, CSO07, and CSR07 have smooth disparity change in the objects thanks to the smoothing. There are few erroneous estimations of texture-less portions such as the tree lattice portion on the upper right. Compared with CSC07 and CSO07, CSR07 shows that the lacks of objects such as cones in front are corrected by the cross recovery.

The result of teddy is shown in **Figure 5**. The roof of the house is texture-less, CNC07 has a lot of matching failures. CSC07, CSO07, and CSR07 use the same center search result as CNC07, but there is no big mis-prediction thanks to the smoothing. Furthermore, in CSO07, the occlusion part between the roof and the back wall is correctly detected by the one-way check. Even without using the cross check, the roof boundary is clear and there is no blur due to erroneous estimation on the boundary.

The accuracy and execution time of each algorithm for each image are shown in **Table 1**. Image resolution for each image is listed in the “res.” row. “#SP L, R” means the number of SPs in the left image and right image. Compared to FNC15, the accuracy degradation of CNC07 is great, but that of CSO07 is little. The processing time of CSO07 is about 1/100 of FNC15. In the proposed method with the one-way check, the center search can be done only in one direction



**Figure 4.** Results of cones.

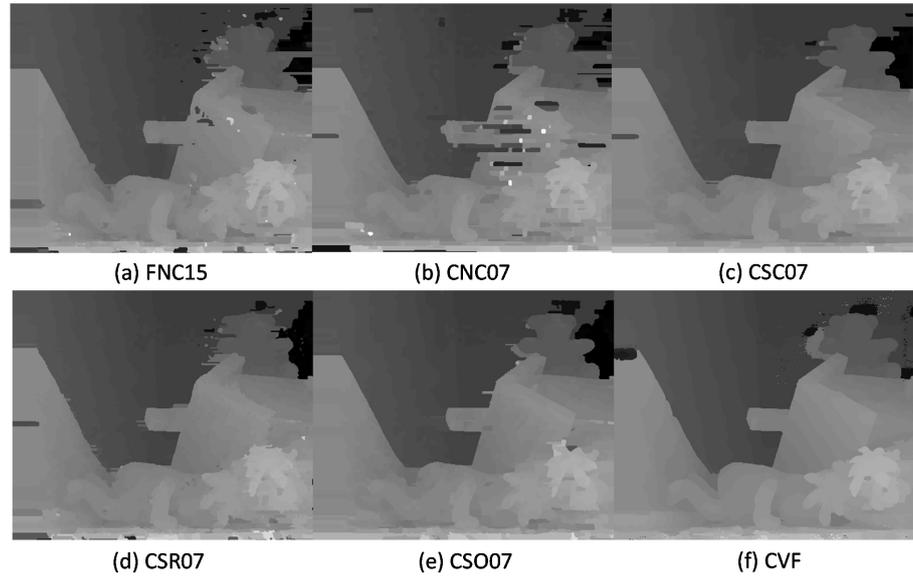


Figure 5. Results of teddy.

Table 1. Accuracy and execution time of each algorithm for each image.

	cones			teddy			tsukuba			venus		
res.	450 × 375			450 × 375			384 × 288			434 × 383		
#SP L, R	3127,3140			3172,3193			2072,2054			3139,3133		
alg.	RMS	BP (%)	Time(ms)	RMS	BP (%)	Time (ms)	RMS	BP (%)	Time (ms)	RMS	BP (%)	Time (ms)
FNC15	5.16	19.24	19858.70	6.27	22.52	22920.66	1.30	5.59	13047.55	1.74	7.53	23945.62
CNC07	5.32	21.38	282.57	6.92	24.27	290.48	1.67	9.02	221.55	2.76	11.62	297.15
CSC07	5.10	17.19	304.41	5.68	19.36	309.20	1.47	8.89	244.12	1.36	7.95	315.33
CSO07	5.24	18.21	189.50	5.79	20.50	185.69	1.37	7.54	152.00	1.51	9.48	189.30
CSR07	5.18	16.75	318.64	6.10	19.01	321.44	1.43	7.33	255.76	1.32	7.22	336.74
CVF	4.79	12.38	6436.13	5.58	13.72	6114.56	0.81	3.54	3848.08	0.51	0.46	5839.37

thanks to the introduction of the one-way check, and the block size can be small as  $7 \times 7$  by introducing the smoothing. Furthermore, since the support area of the right image has no effect in the case of  $7 \times 7$  block, the SP segmentation of the right image is unnecessary. Due to these factors, the proposed method with the one-way check is fast.

The accuracy of CSR07 in many cases exceeded the accuracy of FCN15, and the execution time was about 1/60. The accuracy of CSR07 exceeded the accuracy of CSC07, and the effect of recovery was shown. The accuracy of CSR07 was lower than that of CVF, but the speed was more than 15 times. The proposed method is as accurate as the full search and orders of magnitude faster than the state of the art. Such a property contributes to real-time generation of depth image generation, resolution enhancement, search range expansion, and expands the application scope of the depth image.

The execution time of each process against the image cones is shown in Table 2.

**Table 2.** Execution time of each process for cones.

Process	Time (ms)
SP segmentation	64.64
Similarity calculation	2.79
Merge	0.44
Full search (15 × 15)	9955.29
Center search (7 × 7)	64.06
Cross check	0.74
Cross check and recovery	14.31
One-way check	12.93
Reliability calculation	4.25
Smoothing with cost filter	4.30
Gap filling	20.55

In the one-way check, error calculation is performed for several left pixels matching the same right pixel, and in case of losers, re-matches are performed. Therefore, the execution time of the one-way check is longer than the cross check that only confirms the matching of the left and right disparities. However, since error calculation is a comparison between pixels, execution time is much shorter than the center search which performs block matching. The execution time of two one-way checks is less than half of the execution time of one center search. Then the execution time of CSO07 is about 60% of CNC07.

## 5. Conclusion

In this paper, we proposed a fast method of depth image generation using SPs. The proposed method reduces the computational complexity without severe accuracy degradation by fully utilizing SPs for all of pixel sampling, creation of support area, and disparity improvement. We introduced one-way check that performs error detection using unidirectional disparity obtained by only once center search. We also introduced a cost filter on SP basis. Disparities of the part without texture are improved by the cost filter which aggregates the weighted matching costs of the surrounding SPs. As a result of the experiment, the execution time of the proposed method was less than 1/100 of the full search and the accuracy degradation was slight. We also proposed a new method called cross recovery that restores lacks of objects problematic in SP-based matching. The accuracy of the method using cross recovery exceeded the full search, and the execution time was about 1/60. Speed up while keeping accuracy extends the application range of the depth image. To realize the real-time image processing of high-resolution video by developing a dedicated processor based on the proposed algorithm is a future work.

## Acknowledgements

This study received the furtherance of Grant in Aid for Scientific Research

(15K00227).

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Roszkowski, M. (2012) Stereo Matching with Superpixels. *Proceedings of SPIE 8454, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments*, Wilga, 15 October 2012. <https://doi.org/10.1117/12.2000139>
- [2] Miyama, M. (2017) Fast Stereo Matching with Super-Pixels Using One-Way Check and Score Filter. *Proceeding of the 7th IEEE International Conference on Control Systems, Computing and Engineering*, Penang, 24-26 November 2017, 278-283. <https://doi.org/10.1109/ICCSCCE.2017.8284419>
- [3] Scharstein, D. and Szeliski, R. (2001) A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Proceedings of 2001 IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV)*, Washington DC, 9-10 December 2001, 131-140.
- [4] Zhang, K., Lu, J. and Lafruit, G. (2009) Cross-Based Local Stereo Matching Using Orthogonal Integral Images. *IEEE Transactions on Circuits and Systems for Video Technology*, **19**, 1073-1079. <https://doi.org/10.1109/TCSVT.2009.2020478>
- [5] Rhemann, C., Hosni, A., Bleyer, M., Rother, C. and Gelautz, M. (2011) Fast Cost-Volume Filtering for Visual Correspondence and Beyond. *Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, 20-25 June 2011, 3017-3024. <https://doi.org/10.1109/CVPR.2011.5995372>
- [6] Tan, P. and Monasse, P. (2014) Stereo Disparity through Cost Aggregation with Guided Filter. *IPOLE—Image Processing on Line*, **4**, 252-275. <https://doi.org/10.5201/ipol.2014.78>
- [7] Hirschmuller, H. (2011) Semi-Global Matching—Motivation, Developments and Applications. *Photogrammetric Week*, **11**, 173-184.
- [8] Kolmogorov, V. and Zabih, R. (2001) Computing Visual Correspondence with Occlusions Using Graph Cuts. *Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001*, Vancouver, 7-14 July 2001, 508-515.
- [9] Sun, J., Zheng, N.N. and Shum, H.Y. (2002) Stereo Matching Using Belief Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 787-800. [https://doi.org/10.1007/3-540-47967-8\\_34](https://doi.org/10.1007/3-540-47967-8_34)
- [10] Ho, Y.S. and Jang, W.S. (2012) Occlusion Detection Using Warping and Cross-Checking Constraints for Stereo Matching. In: Jin, J.S., Xu, C.S. and Xu, M., Eds., *The Era of Interactive Media*, Springer, New York, NY, 363-372.
- [11] Baek, E.T. and Ho, Y. (2016) Occlusion and Error Detection for Stereo Matching and Hole-Filling Using Dynamic Programming. *Electronic Imaging*, 1-6. <https://doi.org/10.2352/ISSN.2470-1173.2016.5.SDA-449>
- [12] Sun, J., Li, Y., Kang, S.B. and Shum, H.Y. (2005) Symmetric Stereo Matching for Occlusion Handling. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA, 20-26 June 2005, 399-406.
- [13] Yamaguchi, K., Hazan, T., McAllester, D. and Urtasun, R. (2012) Continuous Mar-

kov Random Fields for Robust Stereo Estimation. *Computer Vision ECCV 2012*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg.

- [14] Baek, E.T. and Ho, Y.S. (2016) Cost Aggregation with Guided Image Filter and Superpixel for Stereo Matching. *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA 2016)*, Jeju, 13-16 December 2016, 1-4.
- [15] JIN, F. and Li, X. (2015) A Dense Depth Estimation Method Using Superpixels. *12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP 2015)*, Chengdu, 18-20 December 2015, 290-294.
- [16] Zhang, Y., Hartley, R., Mashford, J. and Burn, S. (2011) Superpixels, Occlusion and Stereo. *International Conference on Digital Image Computing Techniques and Applications (DICTA 2011)*, Noosa, 6-8 December 2011, 84-91.
- [17] Miyama, M. (2017) FPGA Accelerator for Super-Pixel Segmentation Featuring Clear Detail and Short Boundary. *IEEE Transactions on Image Electronics and Visual Computing*, **5**, 83-91.