

Article

Stereo Matching Methods for Imperfectly Rectified Stereo Images

Phuc Hong Nguyen and Chang Wook Ahn * 

Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Korea; hongphuc@gist.ac.kr

* Correspondence: cwan@gist.ac.kr; Tel.: +82-62-715-2661

Received: 26 March 2019; Accepted: 17 April 2019; Published: 19 April 2019



Abstract: Stereo matching has been under development for decades and is an important process for many applications. Difficulties in stereo matching include textureless regions, occlusion, illumination variation, the fattening effect, and discontinuity. These challenges are effectively solved in recently developed stereo matching algorithms. A new imperfect rectification problem has recently been encountered in stereo matching, and the problem results from the high resolution of stereo images. State-of-the-art stereo matching algorithms fail to exactly reconstruct the depth information using stereo images with imperfect rectification, as the imperfectly rectified image problems are not explicitly taken into account. In this paper, we solve the imperfect rectification problems, and propose matching stereo matching methods that based on absolute differences, square differences, normalized cross correlation, zero-mean normalized cross correlation, and rank and census transforms. Finally, we conduct experiments to evaluate these stereo matching methods using the Middlebury datasets. The experimental results show the proposed stereo matching methods can reduce error rate significantly for stereo images with imperfect rectification.

Keywords: matching cost; stereo correspondence; high resolution

1. Introduction

Stereo matching is an important process in the field of computer vision, the goal of which is to reconstruct three-dimensional (3D) information from a scene with left and right stereo images [1]. Stereo matching algorithms have been commonly applied in medical imaging and 3D imaging systems, such as satellite-based earth and space exploration, autonomous robots, and vehicle and security systems [2]. Stereo matching is a challenging task due to difficulties such as textureless regions, occlusion, illumination variation, the fattening effect, discontinuity, flying snow, sun flare, and rain blur [3,4].

Sparse stereo matching methods typically use feature descriptors, such as scale-invariant feature transform [5] and speeded-up robust features [6], to compute sparse disparity map, where not all pixels have disparity values [7–9]. Sarkis and Diepold [10] introduced an approach to convert sparse disparity map to dense maps. The efficient large-scale stereo matching method (ELAS) [11] operates on rectified input images, such that correspondences are restricted to the same line in both images.

In our work, we solve the different problem, which input stereo images have been rectified, but the rectification operates imperfectly. Unlike ELAS, our proposed method does not assume that correspondences are restricted to the same line in both images. In addition, our proposed method is a dense stereo matching. There is no interpolation step in our proposed method.

Scharstein et al. [12] classified stereo matching algorithms into local and global algorithms, which consist of steps for matching cost computation, cost aggregation, depth map computation, and depth map refinement phases. The matching cost computation step is required for both types

of stereo matching algorithms and is important to the accuracy of the disparity map. The output of the matching cost computation step is a disparity space image \mathbf{C} [12] in which $C_d(\mathbf{p})$ is the matching cost value of a pixel \mathbf{p} in the reference image, e.g., the left image of a stereo pair, and at a disparity hypothesis d .

Local stereo matching algorithms use cost aggregation techniques to locally smooth the matching cost values in \mathbf{C} . Let \mathbf{C}' be the result of applying a cost aggregation technique to \mathbf{C} . From \mathbf{C}' , a disparity value for \mathbf{p} can be obtained by using a winner-takes-all strategy, as follows:

$$\mathbf{D}_E(\mathbf{p}) = \arg \min_d (\mathbf{C}'(\mathbf{p}, d)), \quad (1)$$

where \mathbf{D}_E is an estimated disparity map.

Global stereo algorithms can use global optimization methods, such as graph-cut [13] or belief propagation [14], to minimize the energy function that constrains the smoothness of the disparities between two neighboring pixels. In global stereo matching, the energy function is first defined and is then solved as an energy minimization problem. A disparity map with higher energy is more erroneous, whereas a disparity map with lower energy is more accurate. The typical form of an energy function in stereo matching is

$$E(\mathbf{D}_E) = E_{data}(\mathbf{D}_E) + E_{smooth}(\mathbf{D}_E), \quad (2)$$

where E_{data} is the measurement of the photo consistency which is computed using a matching cost function. E_{smooth} is a measurement of the smoothness, which is defined as follows:

$$E_{smooth}(\mathbf{D}_E) = \sum_{\langle \mathbf{p}, \mathbf{q} \rangle \in \Omega} s(d_{\mathbf{p}}, d_{\mathbf{q}}) \quad (3)$$

and

$$s(d_{\mathbf{p}}, d_{\mathbf{q}}) = \begin{cases} 0 & \text{if } d_{\mathbf{p}} = d_{\mathbf{q}} \\ \Delta & \text{otherwise} \end{cases} \quad (4)$$

where Δ is a predefined penalty value that balances the smoothness and data terms, Ω is the set of neighboring pixels in the reference image, and $s()$ is a smoothness function that gives a penalty if the disparities of two pixels are different. $d_{\mathbf{p}}$ and $d_{\mathbf{q}}$ are disparity values of pixels \mathbf{p} and \mathbf{q} , respectively.

According to Hirschmuller et al. [15], radiometric differences between stereo images are inherent and inevitable even when the images are produced under controlled lighting and exposure conditions. However, advanced stereo matching cost functions [16,17] can operate robustly with stereo images of different intensity transformations. In other words, the radiometric distortion problem in stereo matching can be solved in the matching cost computation step. Textureless regions, discontinuity, and occlusion problems can be solved by cost aggregation or depth map computation processes [18].

The assumption of existing dense stereo matching algorithms is that input stereo images are perfectly rectified such that correspondent pixels between the rectified stereo images have the same y-coordinate values. This assumption is commonly known as the frontal-parallel assumption. However, obtaining perfect rectification for a stereo pair, especially for large stereo images, is currently a challenge [19]. Therefore, when working on stereo images with high resolution, stereo matching algorithms are required to consider this imperfect rectification problem, as the frontal-parallel assumption does not hold true anymore.

A stereo pair, before used as input stereo images for stereo matching algorithms, typically undergoes a rectification process. The rectification process aims for correspondent pixels between stereo images to be located in the same frontal-parallel lines (or epipolar lines). However, according to [19], it is difficult to achieve perfect results with current rectification methods when operating on a stereo pair with high resolution. Correspondent pixels in stereo images with imperfect rectification

may be located in different epipolar lines [19]. This means that correspondent pixels do not satisfy the frontal-parallel assumption that all dense stereo matching algorithms require. The imperfect problem is unavoidable when rectifying high resolution stereo images, even using advanced rectification methods [19]. At the same time, the need for high resolution stereo images is on the rise [18,19]. However, there is a lack of research on imperfect rectification in stereo matching and most previous studies [20–28] are not aware of the problem of high resolution images.

Existing stereo matching methods are dense methods that compute disparity values for each pixel, and most algorithms implicitly or explicitly make an assumption about epipolar geometry that the corresponding pixels locate in the same epipolar line. Currently, only the Middlebury dataset provides stereo images with high resolution and imperfect rectification, and these stereo images are not included in its benchmark. Therefore, existing research only focuses on low and high stereo images with perfect rectification.

In this paper, we propose several novel matching cost using state-of-the-art matching cost for high resolution stereo images. We use the Middlebury dataset [19] to evaluate the proposed matching cost functions in local and global stereo matching frameworks. The testing local stereo matching algorithms include the absolute different (AD)-based window algorithm, squared difference (SD)-based window algorithm, Rank-based window algorithm, Census-based window algorithm, normalized cross correlation (NCC), and zero-mean normalized cross correlation (ZNCC) [29]. According to [15,30], NCC and ZNCC can be considered a local stereo algorithm, so in our experiments, we do not apply the cost aggregation (via a window) for NCC and ZNCC. The testing global stereo matching algorithms include the AD and graph cut (GC) [13], SD and GC, Rank and GC, and Census and GC algorithms.

2. Matching Cost Functions

2.1. Application to Dense Stereo Matching

Existing stereo matching algorithms operate on the perfect rectification assumption that correspondent pixels are frontal-parallel. Therefore, in the matching cost computation, for a pixel \mathbf{p} in the reference image I , candidate pixels \mathbf{p}' in the target image I' have the same y -coordinate values as \mathbf{p} and only differ with regard to the x -coordinate values. However, when working with stereo images with high resolution, the rectification algorithm can operate imperfectly. As a result, correspondent pixels between stereo images may have different y -coordinate values. This means that the frontal-parallel assumption does not hold true in these cases. Therefore, existing stereo matching algorithms tolerate a new problem that is introduced by the imperfect rectification process.

Let $\mathbf{p} = [x_{\mathbf{p}}, y_{\mathbf{p}}]^T$ be a pixel in the reference image I and $\mathbf{p}' = [x_{\mathbf{p}'}, y_{\mathbf{p}'}]^T$ be a pixel in the reference image I' , and $\mathbf{d} = [d, r]^T$ be a disparity value. Without explicitly stating so, we implicitly use the left image as the reference image. The existing stereo matching algorithms work on the frontal-parallel assumption, so the value r is always set to zero. This fixed value $r = 0$ is the main reason that existing stereo matching algorithms work poorly in stereo images with imperfect rectification. The expansion parameter r can change and is in the interval $[-R, R]$ where R is an expansion range. Let M_1 be a matching cost function in a traditional approach. A matching cost value for the pixel \mathbf{p} and a disparity hypothesis d is computed using the frontal-parallel assumption as follows:

$$\mathbf{C}(\mathbf{p}, d) = M_1(\mathbf{p}, d). \quad (5)$$

Here, the function M_1 takes the coordinate of \mathbf{p} in the reference image and the value d . The coordinate of the correspondent pixel \mathbf{p}' in the target image is computed as follows:

$$\mathbf{p}' = [x_{\mathbf{p}} - d, y_{\mathbf{p}}]^T. \quad (6)$$

The frontal-parallel assumption can be described using Equation (6). The pixel \mathbf{p} in the reference image and correspondent pixel \mathbf{p}' in the target image have the same values $y_{\mathbf{p}}$. This means that for each pixel in the reference image, the correspondent pixel in the target image has the same epipolar line.

The imperfect rectification problem is that correspondent pixels between the left and right images can be located in different epipolar lines. Therefore, the setting to look for the correspondent pixel in Equation (6) fails to correctly recover the disparity information because \mathbf{p}' is constrained to be located in the same epipolar line as \mathbf{p} .

To cope with imperfect rectification, a search space for the correspondent pixel \mathbf{p}' requires to include pixels from above and below the considered line $y = y_{\mathbf{p}}$ in the target image. We redesign the setting to obtain a matching cost value as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r M_2(\mathbf{p}, d, r), \quad (7)$$

where M_2 is a matching cost function in the proposed setting, and $r \in \mathbb{N}$ and $r \in [-R, R]$. The function M_2 takes one more input parameter r that determines how much the search space should be expanded. Our idea in Equation (7) is that for each disparity hypothesis d , a matching cost function should consider pixels above and below the pixel $\mathbf{p}' = [x_{\mathbf{p}} - d, y_{\mathbf{p}}]^T$ in the target image, and the most similar pixel is chosen to compute a matching cost value.

In this paper, we apply this proposed setting for matching cost functions, including AD and SD (pixel-wise matching cost functions), census and rank (transform-based matching cost functions), and NCC and ZNCC (window-based matching cost functions).

2.2. Application to Pixel-Wise Matching Cost Functions

In this subsection, two pixel-wise matching cost functions, AD and SD is modified to adapt with high resolution stereo images. The AD and SD matching cost functions compute a matching cost value for the pixel \mathbf{p} and a hypothesis disparity d using the intensities of \mathbf{p} and \mathbf{p}' . We denote the new functions ImpAD and ImpSD, respectively.

2.2.1. ImpAD

The AD matching cost function computes the absolute value of the intensity difference of a pixel pair. An AD matching cost value measures the similarity between two pixels. Matching cost values of AD are computed as follows:

$$AD(\mathbf{p}, d) = |I(x_{\mathbf{p}}, y_{\mathbf{p}}) - I'(x_{\mathbf{p}} - d, y_{\mathbf{p}})|, \quad (8)$$

where $I(x_{\mathbf{p}}, y_{\mathbf{p}}) = I_{\mathbf{p}}$ is the intensity value of \mathbf{p} in the reference image, and $I'(x_{\mathbf{p}} - d, y_{\mathbf{p}}) = I'_{\mathbf{p}'}$ is the intensity value of \mathbf{p}' in the target image.

As a traditional matching cost function, the AD function requires only the estimated disparity information d to determine the correspondent pixel $\mathbf{p}' = [x_{\mathbf{p}} - d, y_{\mathbf{p}}]^T$ in the target image. The resulting value of $AD(\mathbf{p}, d)$ is simply assigned to the disparity space image \mathbf{C} as follows:

$$\mathbf{C}(\mathbf{p}, d) = AD(\mathbf{p}, d). \quad (9)$$

The ImpAD matching cost function requires not only the estimated disparity value d , but also the expansion value r to determine the correspondent pixel $\mathbf{p}'_i = [x_{\mathbf{p}} - d, y_{\mathbf{p}} + r]^T$ in the target image. Here, we denote \mathbf{p}'_i as a correspondent pixel of \mathbf{p} in the proposed setting, which uses both pieces of information d and r . An ImpAD matching cost value is computed as follows:

$$ImpAD(\mathbf{p}, d, r) = |I(x_{\mathbf{p}}, y_{\mathbf{p}}) - I'(x_{\mathbf{p}} - d, y_{\mathbf{p}} + r)|. \quad (10)$$

Here, \mathbf{p}'_i differs \mathbf{p} from values d and r where $d \in [d_{min}, d_{max}]$ and $r \in [-R, R]$. A matching cost value at pixel \mathbf{p} and disparity hypothesis d in \mathbf{C} is computed as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r ImpAD(\mathbf{p}, d, r). \quad (11)$$

Among different matching cost values for different values r , the minimum matching value is selected to assign to \mathbf{C} as a matching cost value for \mathbf{p} and d .

2.2.2. ImpSD

The SD matching cost function computes the square of the absolute value of the intensity difference between two pixels. A SD matching cost values is computed as follows:

$$SD(\mathbf{p}, d) = (I(x_{\mathbf{p}}, y_{\mathbf{p}}) - I'(x_{\mathbf{p}} - d, y_{\mathbf{p}}))^2. \quad (12)$$

Like AD, the SD matching cost function needs only d to compute the correspondent pixel $\mathbf{p}' = [x_{\mathbf{p}} - d, y_{\mathbf{p}}]^T$ in the target image. The resulting value of $SD(\mathbf{p}, d)$ is set to the disparity space image \mathbf{C} as follows:

$$\mathbf{C}(\mathbf{p}, d) = SD(\mathbf{p}, d) \quad (13)$$

The ImpSD matching cost function needs both the estimated disparity value d and the expansion value r to compute the correspondent pixel $\mathbf{p}'_i = [x_{\mathbf{p}} - d, y_{\mathbf{p}} + r]^T$ in the target image. An ImpSD matching cost value is computed as follows:

$$ImpSD(\mathbf{p}, d, r) = (I(x_{\mathbf{p}}, y_{\mathbf{p}}) - I'(x_{\mathbf{p}} - d, y_{\mathbf{p}} + r))^2 \quad (14)$$

A matching cost value at the pixel \mathbf{p} and a disparity hypothesis d in \mathbf{C} is computed as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r ImpSD(\mathbf{p}, d, r) \quad (15)$$

2.3. Application to Transform-Based Matching Cost Functions

We introduce two transform-based matching cost functions, Rank and Census for high resolution images. The Rank and Census matching cost functions do not depend directly on pixel intensities to compute their matching values. The functions first compute the relative order between the anchor pixel (the pixel at the center of a support window) and its neighbors. Therefore, Rank and Census can operate robustly on stereo images under radiometric distortion. We denote ImpRank and ImpCensus for Rank and Census that are modified and are aware of the imperfect rectification problem.

2.3.1. ImpRank

The Rank matching cost function computes the sum of the order relative to the pixel pairs and results in an integer value that describes the local structure of an image patch. The Rank function is computed as follows:

$$Rank(\mathbf{p}, d) = \left| \sum_{\mathbf{q} \in N_{\mathbf{p}}} \delta(\mathbf{p}, \mathbf{q}) - \sum_{\mathbf{q}' \in N_{\mathbf{p}'}} \delta'(\mathbf{p}', \mathbf{q}') \right|, \quad (16)$$

where $N_{\mathbf{p}}$ and $N_{\mathbf{p}'}$ are sets of neighboring pixels of the pixels \mathbf{p} and \mathbf{p}' in the left I and right I' images, respectively. The indicator functions $\delta(\cdot)$ and $\delta'(\cdot)$ are computed as follows:

$$\delta(\mathbf{p}, \mathbf{q}) = \begin{cases} 1 & \text{if } I_{\mathbf{p}} < I_{\mathbf{q}} \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

and

$$\delta'(\mathbf{p}', \mathbf{q}') = \begin{cases} 1 & \text{if } I'_{\mathbf{p}'} < I'_{\mathbf{q}'} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

where $\mathbf{p}' = \mathbf{p} - [d \ 0]^T$.

The Rank matching cost function computes correspondent candidate pixels \mathbf{p} using only d . Therefore, the Rank function fails to find the correct correspondent pixels \mathbf{p}' to measure matching cost values. The ImpRank matching cost function takes into account the imperfect rectification problem and uses the expansion parameter r to further search for the correspondent pixels \mathbf{p}'_i in the target image.

An ImpRank matching cost value is computed as follows:

$$\text{ImpRank}(\mathbf{p}, d) = \left| \sum_{\mathbf{q} \in N_{\mathbf{p}}} \delta(\mathbf{p}, \mathbf{q}) - \sum_{\mathbf{q}'_i \in N_{\mathbf{p}'_i}} \delta'(\mathbf{p}'_i, \mathbf{q}'_i) \right|, \quad (19)$$

A matching cost value at the pixel \mathbf{p} and a disparity hypothesis d in \mathbf{C} is computed as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r \text{ImpRank}(\mathbf{p}, d, r) \quad (20)$$

2.3.2. ImpCensus

The Census matching cost function transforms a local structure of an image patch into a bit string, and use the Hamming distance to measure the similarity between two bit strings. Bit strings are encoded as follows:

$$\tilde{\zeta}_{\mathbf{p}} = \bigotimes_{\mathbf{q} \in N_{\mathbf{p}}} \delta(\mathbf{p}, \mathbf{q}) \quad (21)$$

and

$$\tilde{\zeta}'_{\mathbf{p}'} = \bigotimes_{\mathbf{q}' \in N_{\mathbf{p}'}} \delta'(\mathbf{p}', \mathbf{q}') \quad (22)$$

where $\tilde{\zeta}_{\mathbf{p}}$ and $\tilde{\zeta}'_{\mathbf{p}'}$ are two bit strings for \mathbf{p} and \mathbf{p}' , respectively.

The Census function is computed as follows:

$$\text{Census}(\mathbf{p}, d) = H(\tilde{\zeta}_{\mathbf{p}}, \tilde{\zeta}'_{\mathbf{p}'}) \quad (23)$$

Take into account the imperfect rectification problem, the proposed ImpCensus matching cost function uses the expansion parameter r in its matching cost computation. An ImpCensus matching cost value is computed as follows:

$$\text{ImpCensus}(\mathbf{p}, d, r) = H(\tilde{\zeta}_{\mathbf{p}}, \tilde{\zeta}'_{\mathbf{p}'_i}) \quad (24)$$

A matching cost value at the pixel \mathbf{p} and a disparity hypothesis d in \mathbf{C} is computed as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r \text{ImpCensus}(\mathbf{p}, d, r) \quad (25)$$

2.4. Application to Window-Based Matching Cost Functions

NCC and ZNCC require support windows and use direct intensity values in their matching cost computation. Like SAD, NCC and ZNCC can be considered local stereo matching methods because disparity maps from these two methods have local smoothness of disparity values [15]. NCC and ZNCC can be computed efficiently using box filtering (BF) [31] or integral image (II) [32] techniques.

2.4.1. ImpNCC

NCC can tolerate small brightness changes between stereo images due to locally dividing by the standard deviation. An NCC matching cost value is computed as follows:

$$NCC(\mathbf{p}, d) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_{\mathbf{q}} \times I_{\mathbf{q}'})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} I_{\mathbf{q}}^2 \times \sum_{\mathbf{q}' \in N_{\mathbf{p}'}} I_{\mathbf{q}'}^2}} \quad (26)$$

We denote ImpNCC for NCC that is aware of the imperfect rectification problem. An ImpNCC matching cost values is computed as follows:

$$ImpNCC(\mathbf{p}, d, r) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_{\mathbf{q}} \times I_{\mathbf{q}'})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} I_{\mathbf{q}}^2 \times \sum_{\mathbf{q}' \in N_{\mathbf{p}'}} I_{\mathbf{q}'}^2}} \quad (27)$$

A matching cost value at the pixel \mathbf{p} and a disparity hypothesis d in \mathbf{C} is computed as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r ImpNCC(\mathbf{p}, d, r) \quad (28)$$

2.4.2. ImpZNCC

The brightness of stereo images can vary due to lighting and exposure conditions. The stereo images can be first locally normalized by subtracting the mean and dividing by the standard deviation.

A ZNCC matching cost value is computed as follows:

$$ZNCC(\mathbf{p}, d) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_{\mathbf{q}} - \bar{I}_{\mathbf{p}}) \times (I_{\mathbf{q}'} - \bar{I}_{\mathbf{p}'})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_{\mathbf{q}} - \bar{I}_{\mathbf{p}})^2 \times \sum_{\mathbf{q}' \in N_{\mathbf{p}'}} (I_{\mathbf{q}'} - \bar{I}_{\mathbf{p}'})^2}} \quad (29)$$

We denote ImpZNCC for ZNCC that is aware of the imperfect rectification problem. An ImpZNCC matching cost value is computed as follows:

$$ImpZNCC(\mathbf{p}, d, r) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_{\mathbf{q}} - \bar{I}_{\mathbf{p}}) \times (I_{\mathbf{q}'} - \bar{I}_{\mathbf{p}'})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (I_{\mathbf{q}} - \bar{I}_{\mathbf{p}})^2 \times \sum_{\mathbf{q}' \in N_{\mathbf{p}'}} (I_{\mathbf{q}'} - \bar{I}_{\mathbf{p}'})^2}} \quad (30)$$

A matching cost value at pixel \mathbf{p} and a disparity hypothesis d in \mathbf{C} is computed as follows:

$$\mathbf{C}(\mathbf{p}, d) = \min_r ImpZNCC(\mathbf{p}, d, r) \quad (31)$$

Like ZNCC, ImpZNCC can be computed efficiently by using the BF and II techniques. Let \times and $/$ be element-wise multiplication and division of two matrices, respectively. Algorithm 1 shows the procedure to compute the ImpZNCC matching cost function. Computation of the sum over a support window in Algorithm 1 is computed fast and efficiently using the BF technique. In Algorithm 1, a value at position $[x, y]^T$ in $\mathbf{K}'_{d,r}$ is computed as $\mathbf{K}'_{d,r}(x, y) = \mathbf{K}'(x - d, y + r)$.

Algorithm 1 The procedure of ImpZNCC matching cost function to construct **C**.

Input: Left and right images I and I' , window size W , expansion range R .

1. $\bar{I} \leftarrow$ compute average over W for I using BF
 2. $\bar{I}' \leftarrow$ compute average over W for I' using BF
 3. $\mathbf{K} \leftarrow I - \bar{I}$
 4. $\mathbf{K}' \leftarrow I' - \bar{I}'$
 5. $\mathbf{K}^2 \leftarrow \mathbf{K} \times \mathbf{K}$
 6. $\mathbf{K}'^2 \leftarrow \mathbf{K}' \times \mathbf{K}'$
 7. $\mathbf{D} \leftarrow$ compute sum over W for \mathbf{K}^2 using BF
 8. $\mathbf{D}' \leftarrow$ compute sum over W for \mathbf{K}'^2 using BF
 9. **For** $d = d_{min}$ to d_{max} **do**
 10. **For** $r = -R$ to R **do**
 11. $\mathbf{M} \leftarrow \mathbf{K} \times \mathbf{K}'_{d,r}$
 12. $\mathbf{S} \leftarrow$ compute sum over W from \mathbf{M} using BF
 13. $\mathbf{C}(d) \leftarrow$ compute $\mathbf{S} / \sqrt{\mathbf{D} \times \mathbf{D}'}$
 14. **end for**
 15. **end for**
 16. **Return** **C**
-

3. Experimental Results

We used the Middlebury [19,33] dataset to measure the performance of matching cost functions including AD, SD, NCC, ZNCC, Rank, Census, ImpAD, ImpSD, ImpNCC, ImpZNCC, ImpRank, and ImpCensus in local and global frameworks. In the present experiments, we do not intend to compare the performance of the test matching cost functions and stereo matching algorithms. We plan to compare the performance of stereo matching algorithms before and after applying the modification to solve the imperfect rectification problem.

For each of the test matching cost functions, we implemented local and global stereo matching algorithms that use the function in the matching cost computation. For local stereo matching algorithms, we used a 15×15 window to aggregate matching costs using **C**. For global stereo matching, we use graph-cut (GC) [34] to smooth **C**. We used the source code of GC in [35]. We carefully and optimally choose the parameters of GC for global stereo algorithms, which use AD, SD, Rank, and Census, by using stereo images with perfect rectification conditions as training examples. The global stereo matching algorithms, which are based on ImpAD, ImpSD, ImpRank, and ImpCensus, use the same parameter values as the global algorithms that are based on the AD, SD, Rank, and Census matching cost functions, respectively.

According to [15,30], NCC and ZNCC can be considered local stereo matching algorithms; hence, we do not apply cost aggregation techniques and global optimization methods for NCC, ZNCC, ImpNCC, and ImpZNCC. For the Rank, Census, ImpRank, ImpCensus, NCC, ZNCC, ImpNCC, and ImpZNCC functions, which require a support window, we used the 9×9 window.

For AD, SD, ImpAD, and ImpSD, each pixel of the input stereo images is subtracted by a mean value which is computed by an image window of the pixel. As a result, these four matching cost functions can reduce the effect of illumination different between the stereo images, and we can measure better the effect of the modification that solve the imperfect rectification problem. We used the 9×9 window for this mean subtraction.

The performance of these four matching cost functions is measured by using the winner-takes-all strategy for **C**. All of the matching cost algorithms were evaluated using the average percentage of erroneous pixels in all zones, except occluded areas, and were computed at a 2-pixel error threshold. This error threshold is a default value in Middlebury benchmark 3 [19]. The error percentage (*Err*) was computed as follows:

$$Err(\%) = \frac{100}{|I_{nocc}|} \sum_{\mathbf{p} \in I_{nocc}} \begin{cases} 0, & \text{if } |\mathbf{D}_E(\mathbf{p}) - \mathbf{D}_G(\mathbf{p})| \leq 2 \\ 1, & \text{otherwise,} \end{cases} \quad (32)$$

where I_{nocc} is the set of all nonoccluded pixels, $|I_{nocc}|$ is the number of pixels in I_{nocc} , and $\mathbf{D}_G(\mathbf{p})$ and $\mathbf{D}_E(\mathbf{p})$ are the ground truth and estimated disparity at \mathbf{p} , respectively.

Middlebury dataset 3 [36] provides the test and training stereo images with different conditions: varying illumination and exposure, and both perfect and imperfect rectification problems. The training stereo images are with ground truth, whereas the test datasets are not. The Middlebury benchmark 3 compares submitted stereo matching algorithms using the test dataset with these four conditions. However, in this paper, we focus on solving the imperfect rectification problem of stereo images. Therefore, in our experiments, we use the training datasets, which contain stereo images with imperfect rectification and varying illumination and exposure. Table 1 presents data for the stereo images in the training datasets. We implemented three versions with $R = 0$, $R = 1$, and $R = 2$, respectively. The algorithms with $R = 0$ has no effect on matching cost function. Therefore, for example, ImpZNCC with $R = 0$ is simply ZNCC.

Table 1. Stereo images with imperfect rectification in the Middlebury training datasets of version 3.

Dataset	Height	Width	Disparity	Dataset	Height	Width	Disparity
Adirondack	1984	2872	290	Playroom	1904	2796	330
Backpack	1988	2948	260	Playtable	1852	2720	290
Bicycle	1968	3052	180	Recycle	1944	2880	260
Cable	1916	2816	460	Shelves	1988	2952	240
Classroom	1896	2996	260	Storage	1988	2792	660
Couch	1992	2296	630	Sword1	2004	2928	260
Flowers	1984	2888	640	Sword2	1956	2884	370
Motorcycle	1988	2964	280	Umbrella	2008	2960	250
Pipes	1940	2940	300				

3.1. ImpCensus and ImpRank

We conducted experiments to evaluate the performance of Census, Rank, ImpCensus, and ImpRank matching cost functions in local and global stereo matching approaches. Denote ImpCensus/Win/R1 as a local stereo matching algorithm that uses the ImpCensus matching cost function with $R = 1$ to construct **C** and aggregates matching costs using a window. In addition, denote ImpCensus/GC/R1 as a global stereo matching algorithm that uses ImpCensus with $R = 1$ and GC to globally optimize the energy function, as described in Equation (2). Similarly, other denoted stereo matching algorithms can be used by changing the matching cost functions and the R values.

Figure 1 shows the results of the ImpCensus-based stereo matching algorithms using the Backpack stereo images with different R values. Disparity maps in the second line are the result of the ImpCensus-based local algorithms, whereas the third line shows the disparity maps of the ImpCensus-based global algorithms. Census/Win and Census/GC produced the most erroneous disparity maps because they were un-aware of the imperfect rectification problem. ImpCensus/GC/R1 and ImpCensus/GC/R2 reduced the error rates. The error rate reduction is clearly seen from Figure 1g,h, especially in textured image regions. These observations agree with those in [19] that the imperfect rectification problem commonly happens in textured image regions.

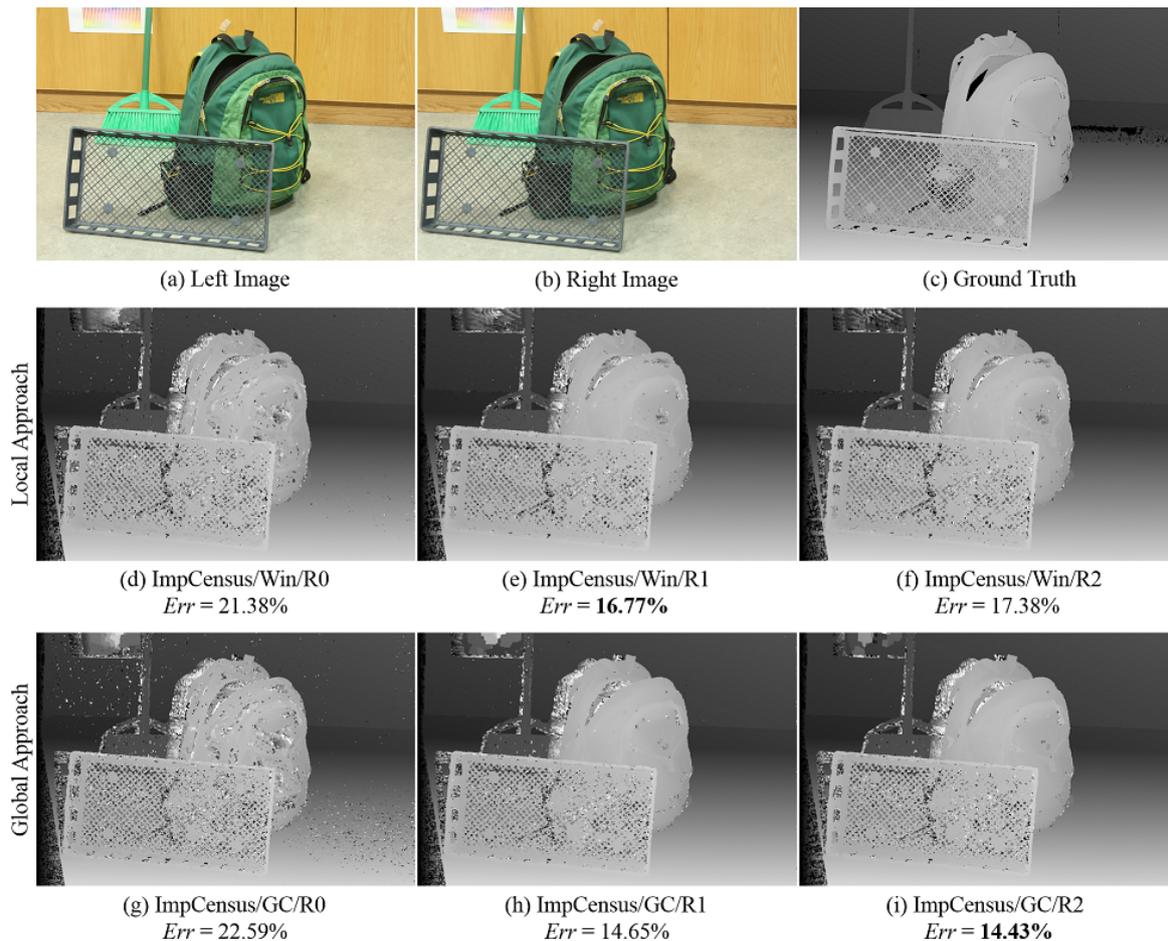


Figure 1. Results of the ImpCensus-based stereo matching algorithms with different R values using the Backpack stereo images with imperfect rectification. (a) Left image. (b) Right image. (c) Ground truth. (d) Disparity map of Census/Win ($Err = 21.38\%$). (e) Disparity map of ImpCensus/Win/R1 ($Err = 16.77\%$). (f) Disparity map of ImpCensus/Win/R2 ($Err = 17.38\%$). (g) Disparity map of Census/GC ($Err = 22.59\%$). (h) Disparity map of ImpCensus/GC/R1 ($Err = 14.65\%$). (i) Disparity map of ImpCensus/GC/R2 ($Err = 14.43\%$).

Tables 2 and 3 show the quantitative results of local and global stereo matching algorithms that use Rank and ImpRank, and Census and ImpCensus, respectively. The ImpCensus-based stereo matching algorithms outperformed the Census-based algorithms for all the test stereo images. Similarly, the performance of the ImpRank-based stereo matching algorithms were superior to the Rank-based algorithms. In the Playtable stereo images, for example, the modification allows the ImpCensus-based local algorithm to reduce the error rate by up to 27.9% (65.28% of Census/Win and 37.38% of ImpCensus/Win/R2). On the other hand, in a global approach, the error rate of ImpCensus/GC/R2 was 39% smaller than that of Census/GC (70.29% of Census/Win and 31.20% of ImpCensus/Win/R2).

Table 2. Error rates of the Census- and ImpCensus-based stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Local Algorithms				Global Algorithms			
Dataset	Census	ImpCensus/R1	ImpCensus/R2	Dataset	Census	ImpCensus/R1	ImpCensus/R2
Adirondack	45.30	38.45	37.62	Adirondack	52.71	37.67	31.14
Backpack	21.38	16.77	17.38	Backpack	22.59	14.65	14.43
Bicycle	51.21	45.10	42.12	Bicycle	53.69	43.13	35.01
Cable	51.84	45.47	42.46	Cable	63.39	47.07	36.37
Classroom	30.41	26.89	28.45	Classroom	38.47	25.49	18.85
Couch	30.65	29.42	30.50	Couch	33.34	28.05	26.83
Flowers	61.64	56.38	54.37	Flowers	64.70	54.43	49.27
Motorcycle	29.50	21.63	20.62	Motorcycle	37.30	21.25	17.52
Pipes	33.59	28.08	25.64	Pipes	38.46	27.52	22.70
Playroom	45.23	43.14	43.64	Playroom	49.79	42.09	39.13
Playtable	65.28	39.50	37.38	Playtable	70.29	34.92	31.20
Recycle	44.81	40.27	40.39	Recycle	50.74	36.91	31.72
Shelves	54.33	48.27	48.11	Shelves	56.88	47.38	44.78
Storage	54.70	49.74	47.33	Storage	63.41	51.99	44.81
Sword1	16.45	15.61	16.43	Sword1	18.41	14.08	13.99
Sword2	70.24	62.97	57.66	Sword2	71.63	52.73	36.92
Umbrella	66.17	63.64	64.40	Umbrella	70.62	62.23	56.46
Average	45.46	39.49	38.50	Average	50.38	37.74	32.42

Table 3. Error rates of the Rank- and ImpRank-based stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Local Algorithms				Global Algorithms			
Dataset	Rank	ImpRank/R1	ImpRank/R2	Dataset	Rank	ImpRank/R1	ImpRank/R2
Adirondack	59.68	50.66	54.20	Adirondack	48.30	34.06	33.27
Backpack	25.60	20.21	24.27	Backpack	20.18	14.64	16.01
Bicycle	60.81	52.77	51.79	Bicycle	48.57	35.77	31.24
Cable	67.54	60.01	63.09	Cable	51.65	34.82	35.74
Classroom	41.49	38.41	48.13	Classroom	23.91	11.23	10.29
Couch	38.48	35.84	45.36	Couch	30.53	27.18	32.48
Flowers	72.03	62.64	62.66	Flowers	60.68	48.51	46.57
Motorcycle	43.21	29.73	31.26	Motorcycle	30.47	17.86	17.55
Pipes	42.67	34.91	34.13	Pipes	32.96	25.24	23.51
Playroom	55.61	50.74	54.73	Playroom	45.05	38.27	39.77
Playtable	73.64	54.26	52.70	Playtable	69.88	39.36	40.49
Recycle	64.45	53.01	56.38	Recycle	43.92	30.58	29.46
Shelves	58.34	53.21	57.39	Shelves	55.57	46.56	46.75
Storage	69.30	62.34	64.56	Storage	57.90	36.67	35.60
Sword1	20.71	19.07	23.50	Sword1	14.08	11.57	13.80
Sword2	78.81	75.55	77.43	Sword2	63.11	44.13	39.95
Umbrella	71.83	70.07	72.47	Umbrella	64.13	52.05	50.66
Average	55.54	48.44	51.41	Average	44.76	32.26	31.95

For the Census- and ImpCensus-based local and global stereo matching algorithms, average error rates of ImpCensus/Win/R1 (39.49%) and ImpCensus/Win/R2 (38.50%) were about 6% smaller than that of Census/Win (45.46%), whereas average error rates of ImpCensus/GC/R1 (37.74%) and ImpCensus/GC/R2 (32.42%) were more than 12% smaller than that of Census/GC (50.38%). Similarly, the awareness of high resolution images had the positive effect for the ImpRank-based stereo matching algorithms such that the ImpRank-based algorithms with $R = 1$ and $R = 2$ had smaller average error rates than the Rank-based algorithms.

3.2. ImpAD and ImpSD

We performed experiments to evaluate the performance of AD, SD, ImpAD, and ImpSD in local and global stereo matching approaches. Tables 4 and 5 show the quantitative results of local

and global stereo matching algorithms that use AD and ImpAD, and SD and ImpSD, respectively. For all of the test stereo images, ImpAD/Win/R1 and ImpAD/Win/R2 outperformed AD/Win, and ImpAD/GC/R1 and ImpAD/GC/R2 were superior to AD/GC. Similarly, the error rates of ImpSD/Win/R1 and ImpSD/Win/R2 were smaller than those of SD/Win, and ImpSD/GC/R1 and ImpSD/GC/R2 performed better than ImpSD/GC/R2 for all the test stereo pairs.

Table 4. Error rates of the AD- and ImpAD-based stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Local Algorithms				Global Algorithms			
Dataset	AD	ImpAD/R1	ImpAD/R2	Dataset	AD	ImpAD/R1	ImpAD/R2
Adirondack	54.17	46.33	47.48	Adirondack	42.09	29.28	31.11
Backpack	28.57	22.54	24.39	Backpack	38.25	21.59	22.28
Bicycle	61.27	55.25	53.39	Bicycle	50.68	38.03	35.97
Cable	65.63	59.14	60.25	Cable	43.09	33.97	33.32
Classroom	40.77	38.37	44.96	Classroom	18.13	12.42	12.96
Couch	37.95	35.09	40.32	Couch	33.86	30.99	33.78
Flowers	69.47	62.56	62.35	Flowers	59.56	52.83	47.05
Motorcycle	41.39	28.82	28.54	Motorcycle	40.81	25.01	22.64
Pipes	43.49	35.38	33.01	Pipes	49.42	30.39	25.62
Playroom	50.55	49.43	51.82	Playroom	43.44	40.26	41.67
Playtable	69.14	50.99	46.05	Playtable	67.20	44.29	35.48
Recycle	56.66	52.55	55.04	Recycle	30.53	25.90	30.18
Shelves	57.14	52.39	54.05	Shelves	55.84	49.21	49.84
Storage	74.25	68.04	70.28	Storage	68.02	49.34	45.84
Sword1	28.13	23.65	26.50	Sword1	34.18	18.98	19.63
Sword2	76.54	73.75	74.28	Sword2	54.70	42.99	36.46
Umbrella	70.67	68.18	69.36	Umbrella	47.41	42.60	43.98
Average	54.46	48.38	49.53	Average	45.72	34.59	33.40

Table 5. Error rates of the SD- and ImpSD-based stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Local Algorithms				Global Algorithms			
Dataset	SD	ImpSD/R1	ImpSD/R2	Dataset	SD	ImpSD/R1	ImpSD/R2
Adirondack	54.19	46.28	48.20	Adirondack	41.67	31.37	37.42
Backpack	29.46	23.01	25.15	Backpack	40.88	23.02	23.36
Bicycle	62.21	56.31	54.57	Bicycle	50.23	44.38	42.52
Cable	65.30	59.24	61.41	Cable	41.96	34.37	35.24
Classroom	40.46	38.15	41.93	Classroom	16.49	12.59	13.18
Couch	38.35	35.92	42.31	Couch	34.86	32.33	36.47
Flowers	70.10	63.60	64.52	Flowers	56.56	44.75	51.13
Motorcycle	42.07	29.45	29.84	Motorcycle	41.33	25.47	25.17
Pipes	44.34	35.90	33.85	Pipes	50.06	30.97	27.12
Playroom	51.04	50.32	52.23	Playroom	42.66	42.39	45.01
Playtable	69.25	51.93	48.25	Playtable	67.74	46.73	37.77
Recycle	56.22	52.67	54.10	Recycle	32.77	30.59	27.42
Shelves	56.72	52.35	54.70	Shelves	53.70	50.51	53.21
Storage	74.35	67.51	70.09	Storage	66.77	46.59	46.97
Sword1	29.97	25.11	27.80	Sword1	36.40	19.40	19.93
Sword2	76.56	73.68	74.99	Sword2	51.32	41.65	38.54
Umbrella	70.34	68.05	69.78	Umbrella	47.54	44.45	42.62
Average	54.76	48.79	50.22	Average	45.47	35.39	35.47

We computed the average performance of each of the test stereo matching algorithms for the test stereo images. For the AD- and ImpAD-based stereo matching algorithms, AD/Win and AD/GC

had the largest errors in their corresponding groups, with average error rates of 54.46% and 45.72%, respectively. In contrast, ImpAD/Win/R1 and ImpAD/GC/R1 had the better performance in the local and global approaches, respectively. ImpAD/Win/R1 performed with average error rates of 48.38%, whereas ImpAD/GC/R1 operated at 34.59% for the test stereo pairs.

For the SD- and ImpSD-based stereo matching algorithms, SD/Win and SD/GC had the largest errors in their correspondent groups, with the average error rates of 54.76% and 45.47%, respectively. In contrast, ImpSD/Win/R1 and ImpSD/GC/R1 had the best performance in the local and global approach, respectively. ImpSD/Win/R1 performed with average error rate of 48.79%, whereas ImpSD/GC/R1 had an error rate of 35.39% over the test stereo pairs.

3.3. ImpNCC and ImpZNCC

We evaluated the performance of NCC and ZNCC with and without using the modification. We evaluated the performance of NCC, ImpNCC, ZNCC, and ImpZNCC directly from the corresponding disparity space image C using a winner-take-all strategy. Denote ImpNCC/R1 as a matching cost function that uses ImpNCC with $R = 1$ to construct C .

Figure 2 shows the results of the ImpZNCC matching cost functions with different R values using the Motorcycle stereo images. Figure 2a,b show the left and right images, whereas the ground truth of the left image is shown in Figure 2c. Disparity maps of ZNCC, ImpZNCC/R1, and ImpZNCC/R2 are shown in Figure 2d–f, respectively. ZNCC produced the most erroneous disparity maps with an average error rate of 49.02% because ZNCC ignores the imperfect rectification problem. ImpZNCC/R1 and ImpZNCC/R2 reduced the error rates with average error rates of 43.73% and 43.64%, respectively.

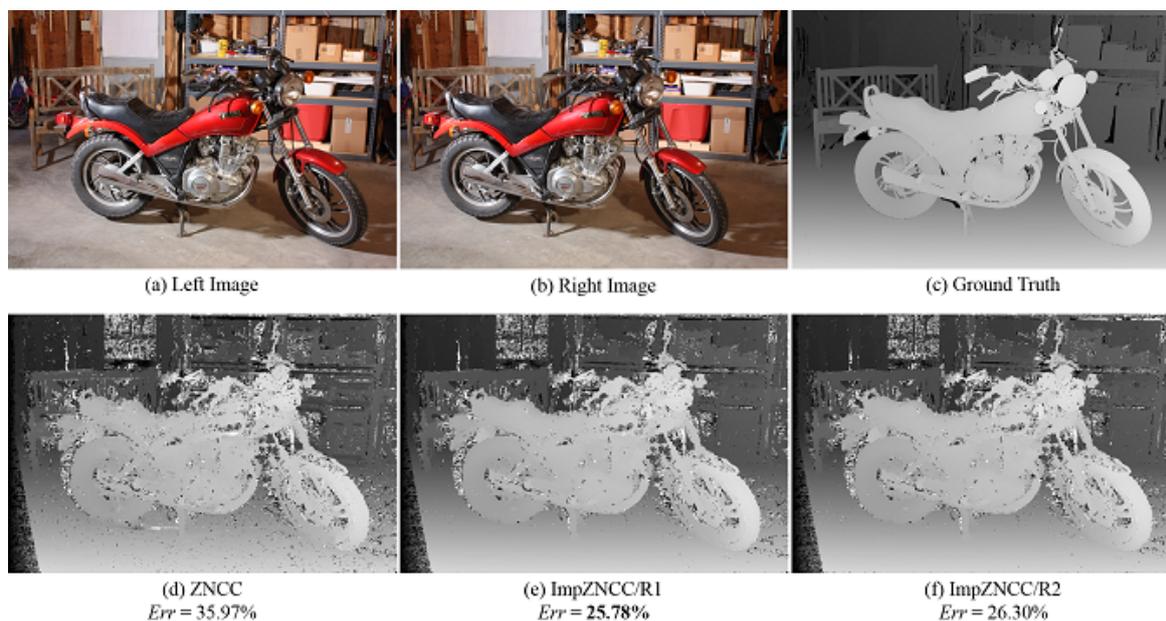


Figure 2. Results of the ImpZNCC-based stereo matching algorithms with different R values using the Motorcycle stereo images with imperfect rectification. (a) Left image. (b) Right image. (c) Ground truth. (d) Disparity map of ZNCC ($Err = 35.97\%$). (e) Disparity map of ImpZNCC/R1 ($Err = 25.78\%$). (f) Disparity map of ImpZNCC/R2 ($Err = 26.30\%$).

Tables 6 and 7 show the quantitative results of the NCC, ImpNCC, ZNCC, and ImpZNCC matching functions, respectively. Using the modification, NCC had the worst performance when producing more erroneous disparity maps than ImpNCC/R1 and ImpNCC/R2. Similarly, the awareness of high resolution images improved the performance of ImpZNCC/R1 and ImpZNCC/R2, which were superior to ZNCC for all of the test stereo pairs.

Table 6. Error rates of the NCC and ImpNCC stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Dataset	NCC	ImpNCC/R1	ImpNCC/R2
Adirondack	49.40	45.22	46.11
Backpack	25.42	21.43	22.11
Bicycle	58.23	54.09	53.12
Cable	54.85	49.64	48.51
Classroom	43.38	41.67	43.37
Couch	35.10	35.34	36.11
Flowers	62.18	59.61	59.31
Motorcycle	36.17	26.90	27.52
Pipes	36.61	30.90	29.05
Playroom	50.22	49.12	48.30
Playtable	66.29	39.50	40.61
Recycle	52.98	52.56	53.83
Shelves	54.60	48.88	49.72
Storage	56.47	53.05	52.93
Sword1	25.85	23.91	24.96
Sword2	74.35	70.78	68.46
Umbrella	72.98	72.26	72.91
Average	50.30	45.58	45.70

Table 7. Error rates of the ZNCC and ImpZNCC stereo matching algorithms using imperfectly rectified stereo images of the Middlebury dataset. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Dataset	ZNCC	ImpZNCC/R1	ImpZNCC/R2
Adirondack	47.49	42.45	43.36
Backpack	24.92	21.05	21.68
Bicycle	56.77	51.68	49.75
Cable	57.76	51.58	50.26
Classroom	34.68	32.73	34.46
Couch	35.62	35.98	36.71
Flowers	63.47	60.16	59.44
Motorcycle	35.97	25.78	26.30
Pipes	38.39	32.07	30.01
Playroom	50.37	49.34	48.65
Playtable	68.12	40.03	40.98
Recycle	51.04	49.45	49.31
Shelves	55.46	49.05	49.89
Storage	55.96	51.23	50.13
Sword1	22.46	21.23	22.03
Sword2	70.68	63.53	58.44
Umbrella	67.68	65.99	67.05
Average	49.23	43.73	43.44

3.4. Stereo Image with Radiometric Distortion

Stereo matching algorithms need to operate robustly on stereo images with radiometric distortion such that they can be used for outdoor applications and road-driving images. In this subsection, we evaluated the performance of stereo matching algorithms that are aware of the high resolution images for stereo images with radiometric distortion and imperfect rectification problems. We used two Middlebury sub-datasets in which one sub-dataset had imperfect rectification and varying exposure and the other sub-dataset had imperfect rectification and varying illumination.

In the present experiments, because Census is one of the most robust matching functions for stereo images with radiometric distortions [15], we use only the ImpCensus-based global stereo matching algorithms. Figure 3 shows the results of Census/GC, ImpCensus/GC/R1, and ImpCensus/GC/R2

using two stereo pairs. The second line shows the disparity maps of the test stereo matching algorithms using a stereo pair (a) and (b) with varying exposure and imperfect rectification, whereas the third line shows the disparity maps using a stereo pair (a) and (c) with varying illumination and imperfect rectification. The error rates of ImpCensus/GC/R1 and ImpCensus/GC/R2 were smaller than those of Census/GC in the two stereo pairs.

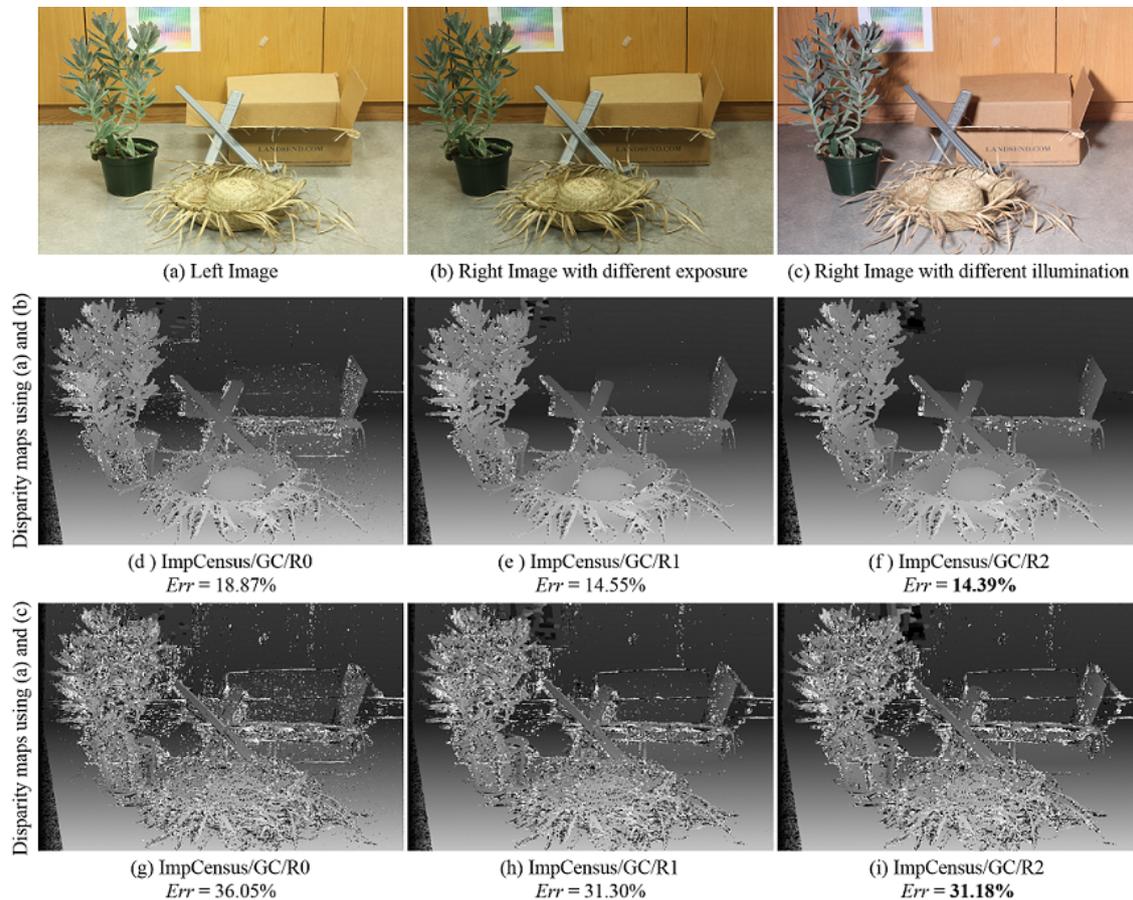


Figure 3. Results of the Census-based stereo matching algorithms using the Sword1 stereo images of imperfect rectification and radiometric distortion. (a) Left image. (b) Right image with varying exposure. (c) Right image with varying illumination. (d–f) Disparity maps using the stereo pair (a,b). (d) Disparity map of Census/GC ($Err = 18.87\%$). (e) Disparity map of ImpCensus/GC/R1 ($Err = 14.55\%$). (f) Disparity map of ImpCensus/GC/R2 ($Err = 14.39\%$). (g–i) Disparity maps using the stereo pair (a,c). (g) Disparity map of Census/GC ($Err = 36.05\%$). (h) Disparity map of ImpCensus/GC/R1 ($Err = 31.30\%$). (i) Disparity map of ImpCensus/GC/R2 ($Err = 31.18\%$).

Tables 8 and 9 show the quantitative results of the local and global stereo matching algorithms, which use ImpCensus and the two Middlebury sub-datasets. For all of the cases in the two tables, the performance of the ImpCensus-based global stereo matching algorithms were improved. Stereo images with varying illumination are often more challenging for stereo matching algorithms than stereo images with varying exposure [15]. Overall, the performance of ImpCensus/GC/R1 and ImpCensus/GC/R2 were superior to Census/GC for all the test stereo images.

Table 8. Error rates of the Census- and ImpCensus-based stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. The stereo images have varying exposure. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Local Algorithms				Global Algorithms			
Dataset	Census	ImpCensus/R1	ImpCensus/R2	Dataset	Census	ImpCensus/R1	ImpCensus/R2
Adirondack	43.80	36.95	36.28	Adirondack	51.66	36.33	28.32
Backpack	20.21	17.06	17.68	Backpack	21.19	15.21	14.28
Bicycle	50.25	44.37	41.46	Bicycle	54.02	42.58	34.16
Cable	49.86	43.36	40.83	Cable	63.30	46.01	36.62
Classroom	38.90	35.97	37.80	Classroom	47.51	37.12	31.62
Couch	34.50	33.04	34.40	Couch	39.27	32.39	30.87
Flowers	63.34	58.66	56.90	Flowers	67.66	58.05	52.68
Motorcycle	27.28	22.60	22.76	Motorcycle	32.79	22.39	19.43
Pipes	34.02	28.30	25.90	Pipes	39.63	27.91	23.09
Playroom	45.01	43.02	43.71	Playroom	49.19	42.17	39.71
Playtable	66.23	40.58	39.07	Playtable	70.89	36.36	33.73
Recycle	48.07	43.61	44.49	Recycle	55.71	41.90	36.92
Shelves	54.89	47.91	47.59	Shelves	57.75	47.69	44.95
Storage	54.14	49.07	46.77	Storage	62.68	50.95	44.50
Sword1	17.06	16.27	17.09	Sword1	18.87	14.55	14.39
Sword2	81.65	76.48	71.25	Sword2	83.43	71.80	56.88
Umbrella	65.12	64.45	65.98	Umbrella	68.53	63.52	59.46
Average	46.72	41.28	40.59	Average	52.01	40.41	35.39

Table 9. Error rates of the Census- and ImpCensus-based stereo matching algorithms using the imperfectly rectified stereo images of the Middlebury dataset. There stereo images have varying illumination. Bold results represent the lowest error rates among the testing methods for each sub-dataset.

Local Algorithms				Global Algorithms			
Dataset	Census	ImpCensus/R1	ImpCensus/R2	Dataset	Census	ImpCensus/R1	ImpCensus/R2
Adirondack	68.60	64.29	64.45	Adirondack	75.65	66.42	63.04
Backpack	33.48	33.20	32.69	Backpack	36.01	33.10	33.18
Bicycle	72.53	71.13	70.62	Bicycle	74.99	71.79	70.45
Cable	82.68	80.65	80.68	Cable	87.24	82.38	79.88
Classroom	73.69	73.43	75.44	Classroom	79.23	76.70	75.19
Couch	53.99	51.60	51.59	Couch	62.34	53.24	48.40
Flowers	76.94	74.54	74.20	Flowers	79.06	74.36	72.13
Motorcycle	48.51	46.76	47.90	Motorcycle	55.12	48.85	46.40
Pipes	58.23	52.11	50.86	Pipes	70.17	58.28	52.28
Playroom	60.61	59.72	60.51	Playroom	65.69	61.09	59.19
Playtable	80.49	76.55	69.63	Playtable	83.04	76.80	62.73
Recycle	62.50	59.45	60.03	Recycle	69.56	60.18	56.58
Shelves	66.01	63.44	64.48	Shelves	69.55	63.29	62.24
Storage	72.36	70.28	69.48	Storage	78.01	72.96	68.94
Sword1	30.36	30.06	31.84	Sword1	36.05	31.30	31.18
Sword2	79.17	75.41	73.05	Sword2	81.05	71.01	59.94
Umbrella	78.88	78.98	79.64	Umbrella	81.43	79.58	79.01
Average	64.65	62.45	62.18	Average	69.66	63.61	60.04

3.5. Using Normal Stereo Images

In this subsection, we evaluated the performance of the proposed stereo matching methods using normal stereo images. In other words, we measure the Imperfect-based method using perfectly rectified Middlebury stereo datasets.

We used sub-datasets, including Aloe, Baby1, Baby2, Baby3, Cloth1, Cloth2, Cloth3, Cloth4, Rocks1, Rocks2, Wood1, and Wood2, to evaluate the Imperfect-based method with different R . Figure 4 shows the quantitative results of the ImpCensus-based method for the Aloe, Baby1, Rock1, and Wood2 image pairs. The ImpCensus-based method explores correspondences in larger searching spaces in terms of the expansion parameter r . As a result, the ImpCensus-based method marginally degraded for perfectly rectified stereo images.

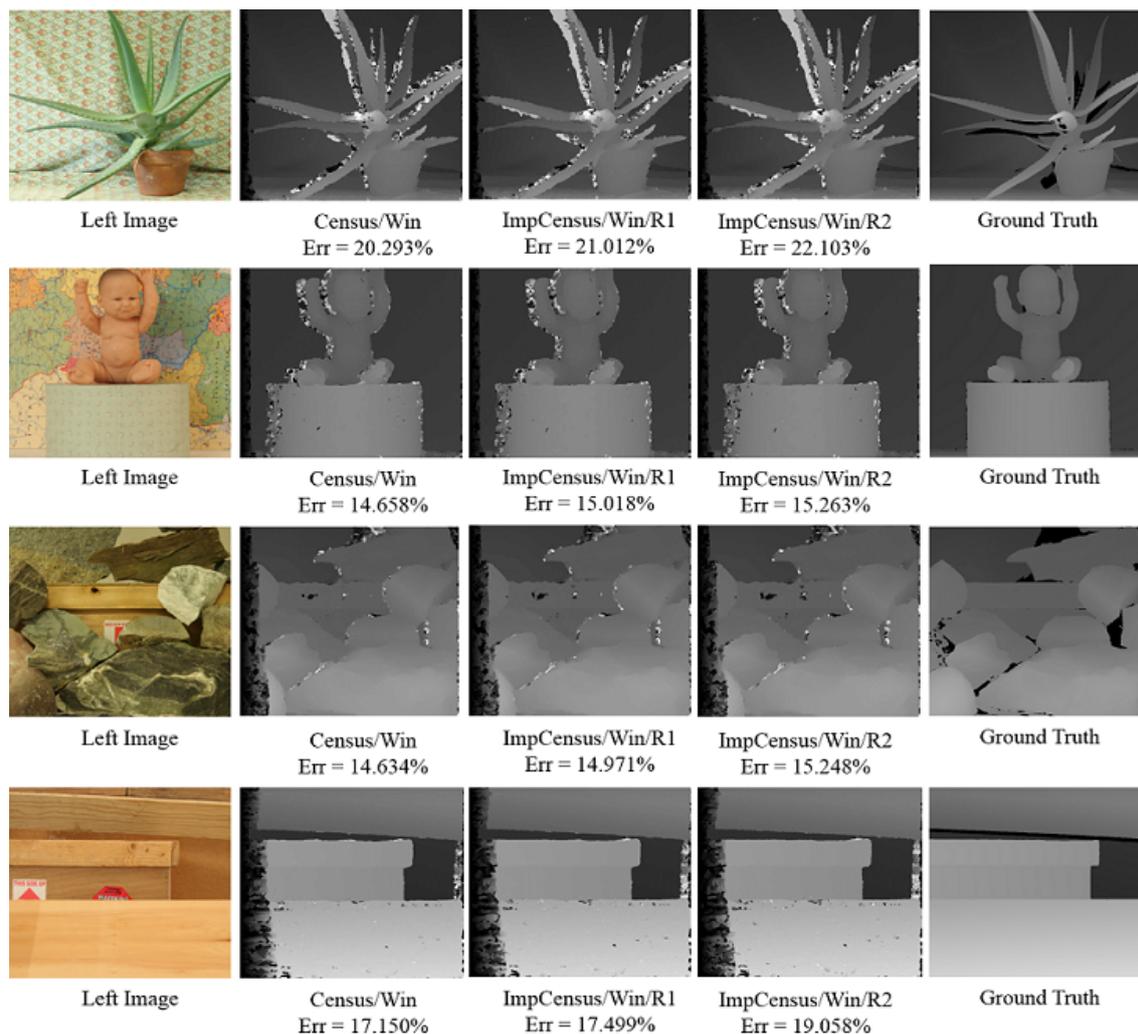


Figure 4. Results of the ImpCensus-based stereo matching algorithms with different R values using the perfectly rectified images. The first column is left images, and the second column is disparity maps for Census/Win. The next two columns are disparity maps for ImpCensus/Win/R1 and ImpCensus/Win/R2, respectively. The last column is ground truths.

Table 10 shows the error rates for the ImpCensus-based method using perfectly rectified stereo images. Clearly, the expansion parameter r had no benefit for these images. Looking for correspondences for larger searching space (with $R = 1$ and $R = 2$) made the ImpCensus-based method more erroneous.

Table 10. Average error rates of proposed local stereo matching algorithms with different R using the perfectly rectified stereo images of the Middlebury dataset.

Dataset	Census/Win	ImpCensus/Win/R1	ImpCensus/Win/R2
Aloe	20.293	21.012	22.103
Baby1	14.658	15.018	15.263
Baby2	20.262	20.879	22.654
Baby3	20.523	20.880	21.835
Bowling1	29.245	30.183	33.428
Bowling2	23.512	24.401	25.628
Cloth1	10.917	11.048	12.553
Cloth2	18.245	18.603	19.083
Cloth3	13.793	14.132	15.834
Cloth4	18.586	18.952	19.463
Flowerpots	26.919	27.802	28.128
Lampshade1	35.201	36.254	38.236
Lampshade2	37.060	37.974	39.137
Midd1	52.165	52.680	53.572
Midd2	49.183	49.800	50.178
Monopoly	35.374	35.967	37.907
Plastic	62.287	62.492	67.283
Rocks1	14.634	14.971	15.248
Rocks2	14.426	14.639	14.817
Wood1	18.174	18.532	19.565
Wood2	17.150	17.499	19.058
Average	26.315	26.844	27.027

3.6. Computation Time

In order to measure the computation times of the matching cost functions, we used the Bicycle stereo images with a resolution of 1968×3052 and a disparity range of 180. We experimentally investigated the matching cost functions, including ImpCensus, ImpRank, ImpAD, ImpSD, ImpNCC, and ImpZNCC, with $R = 0$, $R = 1$, and $R = 2$, respectively. The experimental PC platform had a configuration consisting of an Intel core i7, a 4.00 GHz CPU, and 16.00 GB of memory. Table 11 shows the computation times that are needed for the test matching cost functions to compute the disparity space image C. The testing algorithms requires more computation time when the expansion factor R increases.

Table 11. Computation time (in second) required to compute the disparity space image C.

Function	ImpAD	ImpSD	ImpNCC	ImpZNCC	ImpRank	ImpCensus
$R = 0$	2	2	10	10	4	163
$R = 1$	10	9	32	31	14	485
$R = 2$	14	13	56	55	20	784

As shown in the above tables, methods with the expansion range $R = 1$ clearly reduce the error rates of their original versions. However, methods with $R = 2$ performed comparable or marginally better than those with $R = 1$.

In addition, we further evaluated performance of the proposed local stereo matching methods for $R = 3$ and $R = 4$ using the imperfectly rectified stereo images of the Middlebury dataset, as shown in Table 12. Increasing value for the parameter range R had the negative effects and increase error rates. Therefore, generally, $R = 1$ shows to be the best appropriate value.

Let $|I|$ be image size and D be disparity range. AD and SD are pixel-wise method, so their computational complexities are $\mathcal{O}(|I| \times D)$. Rank and Census are window-based cost functions that each matching cost is computed for windows W . For each window pairs, Rank accumulates values of relative order between center pixel and its neighbors. Therefore, Rank computational complexity is $\mathcal{O}(|I| \times D \times (P - 1))$. Census encodes $(P - 1)$ relative orders into a bit string and then compute

a matching cost by comparing differences between two strings. Therefore, Census computational complexity is $\mathcal{O}(|I| \times D \times P^2)$.

The proposed cost functions with the parameter range R requires to process $K = R \times 2 + 1$ pixels in the right images for each pixel in the left image. Therefore, the computational complexities for ImpAD and ImpSD are $\mathcal{O}(|I| \times D \times K)$, and for ImpCensus and ImpRank are $\mathcal{O}(|I| \times D \times (P - 1) \times K)$.

Table 12. Average error rates of proposed local stereo matching algorithms with different R using the imperfectly rectified stereo images of the Middlebury dataset.

Dataset	R = 0	R = 1	R = 2	R = 3	R = 4
ImpCensus-based	45.46	39.49	38.50	39.27	40.34
ImpRank-based	55.54	48.44	51.41	53.78	56.07
ImpAD-based	54.46	48.38	49.53	49.87	50.62
ImpSD-based	54.76	48.79	50.22	51.13	53.49
ImpNCC-based	50.30	45.58	45.70	45.92	46.53
ImpZNCC-based	49.23	43.73	43.44	44.06	45.33

4. Conclusions

In this paper, we applied the modification to the state-of-the-art stereo matching methods in order to overcome imperfect rectification. We conducted experiments to evaluate these stereo matching methods using the Middlebury datasets. The experimental results indicate that the proposed stereo matching methods largely improved their performance. The proposed stereo matching methods in this paper increases the computation cost for a stereo matching algorithm. To reduce the computation cost or to develop a different approach that can solve the imperfect rectification problem without increasing computation cost is left to our future work.

Author Contributions: Both authors contributed equally to this work and have read and approved the final manuscript.

Funding: This work was supported by the NRF grant funded by the Korea government (MSIT) (NRF-2018R1D1A1A09084148) and (NRF-2018R1D1A1B07049682).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Trucco, E.; Verri, A. *Introductory Techniques For 3-D Computer Vision*; Prentice Hall PTR: Upper Saddle River, NJ, USA, 1998.
2. Cyganek, B.; Siebert, J.P. *Introduction to 3D Computer Vision Techniques and Algorithms*; Wiley: Hoboken, NJ, USA, 2009.
3. Meister, S.; Jähne, B.; Kondermann, D. Outdoor stereo camera system for the generation of real-world benchmark data sets. *Opt. Eng.* **2012**, *51*, 021107. [[CrossRef](#)]
4. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
5. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
6. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. In Proceedings of the Ninth European Conference on Computer Vision, Graz, Austria, 7–13 May 2006.
7. Medioni, G.; Nevatia, R. Segment-based stereo matching. *Comput. Vis. Graph. Image Process.* **1985**, *31*, 2–18. [[CrossRef](#)]
8. Robert, L.; Faugeras, O. Curve-based stereo: Figural continuity and curvature. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Lahaina, HI, USA, 3–6 June 1991.
9. Olson, C.F. Subpixel localization and uncertainty estimation using occupancy grids. In Proceedings of the IEEE International Conference on Robotics and Automation, Detroit, MI, USA, 10–15 May 1999.

10. Sarkis, M.; Diepold, K. Sparse stereo matching using belief propagation. In Proceedings of the IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008.
11. Geiger, A.; Roser, M.; Urtasun, R. Efficient Large-Scale Stereo Matching. In Proceedings of the Asian Conference on Computer Vision (ACCV), Queenstown, New Zealand, 8–12 November 2010.
12. Scharstein, D.; Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* **2002**, *47*, 7–42. [[CrossRef](#)]
13. Kolmogorov, V.; Zabih, R.R. Computing Visual Correspondence with Occlusions using Graph Cuts. *Proc. ICCV* **2001**, *2*, 508–515.
14. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient Belief Propagation for Early Vision. *Int. J. Comput. Vis.* **2006**, *70*, 41–54. [[CrossRef](#)]
15. Hirschmuller, H.; Scharstein, D. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 1582–1599. [[CrossRef](#)] [[PubMed](#)]
16. Nguyen, V.D.; Nguyen, D.D.; Nguyen, T.T.; Dinh, V.Q.; Jeon, J.W. Support local pattern and its application to disparity improvement and texture classification. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 263–276. [[CrossRef](#)]
17. Heo, Y.S.; Lee, K.M.; Lee, S.U. Robust Stereo Matching Using Adaptive Normalized Cross-Correlation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 807–822. [[PubMed](#)]
18. Hosni, A.; Rhemann, C.; Bleyer, M.; Rother, C.; Gelautz, M. Fast Cost-Volume Filtering for Visual Correspondence and Beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 504–511. [[CrossRef](#)] [[PubMed](#)]
19. Scharstein, D.; Hirschmüller, H.; Kitajima, Y.; Krathwohl, G.; Nescic, N.; Wang, X.; Westling, P. High-resolution stereo datasets with subpixel-accurate ground truth. *Conf. Pattern Recognit.* **2014**, *3*. [[CrossRef](#)]
20. Wang, Y.; Wang, K.; Dunn, E.; Frahm, J.-M. Stereo under sequential optimal sampling: A statistical analysis framework for search space reduction. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 485–492.
21. Luo, W.; Schwing, A.G.; Urtasun, R. Efficient Deep Learning for Stereo Matching. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5695–5703.
22. Kowalczyk, J.; Psota, E.T.; Perez, L.C. Real-Time Stereo Matching on CUDA Using an Iterative Refinement Method for Adaptive Support-Weight Correspondences. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 94–104. [[CrossRef](#)]
23. Hirschmüller, H.; Innocent, P.R.; Garibaldi, J. Real-Time Correlation-Based Stereo Vision with Reduced Border Errors. *Int. J. Comput. Vis.* **2002**, *47*, 229–246. [[CrossRef](#)]
24. Li, L.; Zhang, S.; Yu, X.; Zhang, L. PMSC: PatchMatch-Based Superpixel Cut for Accurate Stereo Matching. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 679–692. [[CrossRef](#)]
25. Psota, E.T.; Kowalczyk, J.; Mittek, M.; Perez, L.C. MAP Disparity Estimation Using Hidden Markov Trees. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 2219–2227.
26. Li, L.; Yu, X.; Zhang, S.; Zhao, X.; Zhang, L. 3D Cost Aggregation with Multiple Minimum Spanning Trees for Stereo Matching. 2017. Available online: <http://ao.osa.org/abstract.cfm?URI=ao-56-12-3411> (accessed on 19 April 2019).
27. Kim, K.R.; Kim, C.S. Adaptive smoothness constraints for efficient stereo matching using texture and edge information. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3429–3433.
28. Nahar, S.; Joshi, M.V. A learned sparseness and IGMRF-based regularization framework for dense disparity estimation using unsupervised feature learning. *IPSJ Trans. Comput. Vis. Appl.* **2017**, *9*, 3429–3433. [[CrossRef](#)]
29. Kanade T.; Kano H.; Kimura S.; Yoshida A.; Oda, K. Development of a video-rate stereo machine. In Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, PA, USA, 5–9 August 1995.
30. Antunes, M.; Barreto, J.P. SymStereo: Stereo Matching using Induced Symmetry. *Int. J. Comput. Vis.* **2014**, *109*, 187–208. [[CrossRef](#)]
31. McDonnell, M.J. Box-Filtering techniques. *Comput. Graph. Image Process.* **1981**, *17*, 65–70. [[CrossRef](#)]
32. Crow, F. Summed-area tables for texture mapping. *SIGGRAPH* **1984**, *18*, 207–212. [[CrossRef](#)]

33. Scharstein, D.; Szeliski, R. Middlebury Online Stereo Evaluation. 2002. Available online: <http://vision.middlebury.edu/stereo> (accessed on 19 April 2019).
34. Boykov, Y.; Kolmogorov, V. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [[CrossRef](#)] [[PubMed](#)]
35. Kolmogorov, V. Min-Cut/Max-Flow Algorithm Source Code. 2004. Available online: <http://pub.ist.ac.at/vnk/software.html> (accessed on 19 April 2019).
36. Scharstein, D.; Szeliski, R. Middlebury Online Stereo Evaluation. Available online: <http://vision.middlebury.edu/stereo/eval3> (accessed on 19 April 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).