OCCLUSION HANDLING BASED ON SUPPORT AND DECISION

Dongbo Min, Sehoon Yea and Anthony Vetro

Mitsubishi Electric Research Laboratories (MERL), Cambridge, USA

ABSTRACT

This paper proposes a novel method for handling occluded pixels in stereo images based on a probabilistic voting framework that utilizes a novel support-and-decision process. Occlusion handling aims to assign a reasonable disparity value to occluded pixels in the disparity maps. In an initial step, disparities and their corresponding supports at the occluded pixels are calculated using a probabilistic voting method using the disparities at visible pixels. In this way, the visible pixel information is propagated when the disparities and supports at the occluded pixels are computed. The final disparities for occluded pixels are then computed through an iterative support-and-decision process to propagate the information inside the occluded pixel region. An acceleration technique is also proposed to improve the performance of the iterative support-and-decision process. Experimental results show that the proposed occlusion handling method works well for several challenging stereo images.

***Index Terms*—** Stereo matching, occlusion handling, support-and-decision process, probabilistic voting framework

1. INTRODUCTION

Occlusion is an important and challenging research issue in computer vision and 3D image processing, especially with regards to correspondence problems in stereo and motion analysis [1]. For stereo images, occluded pixels are usually only visible in one image so it is impossible to estimate the corresponding points in the other image. However, reasonable disparity values need to be assigned to the occluded pixels to ensure high-quality image-based rendering and 3D modeling.

To handle occlusions in stereo images, there exist various constraints that could be leveraged based on the 3D geometry. For instance, dynamic programming has been used to estimate disparity and occlusion maps by utilizing the ordering constraint [2]. This approach is very efficient but the ordering constraint is not valid when an image has a thin object. Many approaches have been proposed to estimate the disparity of the occluded pixels by combining the uniqueness constraint into a global optimization method and assigning predefined penalty to the occluded pixels [3, 4]. The disparities of occluded pixels have also been estimated by extrapolation, where the disparities of visible pixels are simply extended out into the occluded region. However, this method is not reliable since the geometric order of occluded pixels may be not valid due to errors in the disparity map.

There are also segmentation-based occlusion handling method that assign the disparities to occluded pixels by performing a plane-fitting with disparities of visible pixels in partly visible segments [5]. Min and Sohn combined the occlusion problem into the cost aggregation scheme using a weighted least squares solution [6]. Only the left disparity field is used to detect occluded pixels in this work. However, for sequential occlusion handling, the geometric order of occlusion is utilized. One drawback of this scheme is that the computational complexity may be high since the method performs a sequential nonlinear filtering on each cost domain repeatedly.

In this paper, we propose a novel approach for handling occlusions in stereo images. We define a support function for occluded pixels in a probabilistic voting framework. This function represents the likelihood that the occluded pixels have a specific disparity value, which can be computed using the disparities of neighboring visible pixels in a probabilistic voting framework.

The remainder of this paper is organized as follows. In Section 2, we introduce the support-and-decision process, and then explain the acceleration method in Section 3. We present the experimental results and conclusion in Section 4 and 5, respectively.

2. PROPOSED OCCLUSION HANDLING

2.1. Problem statement

Given left and right disparity maps, an occlusion map can be computed using a cross-checking method. In this paper, our focus is on the method to assign reasonable disparity values to the occluded pixels, i.e., occlusion detection is not specifically addressed. A number of algorithms have been proposed to assign disparity values to the occluded pixels, e.g., using global optimization or by applying geometric constraints that occluded pixels are likely to have similar disparity values as visible pixels in the background. In this paper, we define a support function for handling occlusion in a probabilistic voting framework. This function represents the likelihood that an occluded pixel has a specific disparity value. The proposed support function $S(m, d)$ is defined as follows:

$$S(m, d) \propto \sum_{n \in \mathrm{N}(m)} p(m, n) f(d, d(n)) o(n) \tag{1}$$

$$f(d, d(n)) = \begin{cases} 1 & if\ d = d(n) \\ 0 & otherwise \end{cases} \qquad o(n) = \begin{cases} 1 & visible \\ 0 & otherwise \end{cases}$$

where $p(m, n)$ represents the probability that pixels m and n have the same disparity value. N is a set of neighboring pixels which are used when computing the support function, and $o(n)$ is a visibility function. Note that the support function is calculated for occluded pixels only, i.e., when $o(m) = 0$ in Eq. (1).

Fig. 1 shows the process that calculates the support at pixel m using the disparities of the visible pixels and their probability values. For each occluded pixel, neighboring visible pixels vote for the disparity candidate which corresponds to their disparity values $(d(n))$ at the occluded pixel m proportional to the probability that the occluded and visible pixels have the same disparity value. In other words, an occluded pixel with have a larger support when there are more neighboring visible pixels with high probability.

The occlusion handling process can be divided into two parts as shown in Fig. 2: initial and iterative support-and-decision processes. In the initial step, the disparities and corresponding support of each occluded pixel is computed using disparities of neighboring visible

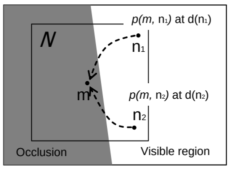


Fig. 1. Computation of support function in probabilistic framework.

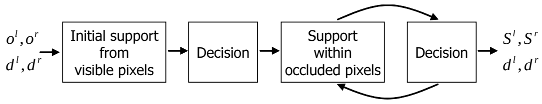


Fig. 2. Overall process of the proposed occlusion handling.

pixels according to Eq. (1). In this way, the visible pixel information is propagated into the occluded region. Building from the disparities and supports established in the initial step, the iterative support-and-decision process is then performed to decide the final disparities and corresponding support of each occluded pixel. Note that the subsequent iterative process is similar to the initial process, except that it uses its own support for occluded pixels and performs a normalization of the support value for each pixel. In this paper, since the occlusion handling step is the same for the left and right disparity maps, we describe the proposed method for only one image.

2.2. Initial support-and-decision process

In order to define the probability function $p(m, n)$ between visible and occluded pixels, we assume that depth discontinuities are likely to correspond to color discontinuities, i.e., the neighboring pixels with similar color usually have similar disparity values. This assumption has been widely used in correspondence problems such as stereo matching and motion estimation. Accordingly, we define a weighting function w that is proportional to the probability function.

$$p(m, n) \propto w(m, n) = \exp\left(-\frac{D(m, n)}{\sigma_S^2} - \frac{D(I_m, I_n)}{\sigma_I^2}\right) \quad (2)$$

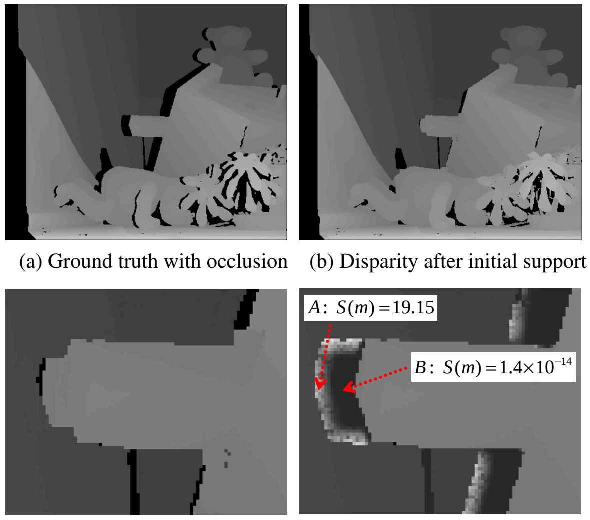
$w(m, n)$ is Gaussian distance function which consists of both geometric and color distances. $D(a, b)$ represents the squared Euclidean distance, and I_m is a 3-dimensional color vector with RGB components. σ_S^2 and σ_I^2 are weighting constants for the geometric and color distances, respectively. Using Eq. (2), the initial support function S_{INI} can be defined as follows:

$$S_{INI}(m, d) = \sum_{n \in \mathrm{N}(m)} w(m, n) f(d, d(n)) o(n) \quad (3)$$

As previously mentioned, the initial support function S_{INI} is computed using the disparities of visible pixels. The weighting function w which is proportional to the probability function helps to localize discontinuities, similar to bilateral and anisotropic diffusion filtering. For accurate occlusion handling, it is also required to gather sufficient support from visible pixels, in order to decide an initial disparity and support for each occluded pixel. Using the initial support function S_{INI}, the initial disparities and the corresponding supports can be computed as follows:

$$\begin{aligned} d(m) &= \arg\max_d S_{INI}(m, d) \\ S(m) &= S_{INI}(m, d(m)) \end{aligned} \quad (4)$$

The support function S represents the reliability of the initial disparity computed at occluded pixel locations. Since the initial support function S_{INI} is computed using the disparity of neighboring



(a) Ground truth with occlusion (b) Disparity after initial support

(c) Cropped image of (b) (d) Support value of (c)

Fig. 3. Disparity map and support values computed using initial support process

visible pixels, the maximum value S of the initial support function represents the likelihood that the occluded pixel m has disparity $d(m)$. Fig. 3 shows an initial disparity map and its support function. It is evident from this example that the support value of pixel A is very large compared to that of pixel B. This implies that the disparity value of pixel A is expected to be sufficiently reliable since the occluded pixel A was supported by neighboring visible pixels with high probability. The support values of occluded pixels determine the reliability of the initial disparities at occluded pixels. The final disparity and the corresponding support values are then calculated by propagating the disparity and support values between the occluded pixels in the iterative support-and-decision process.

2.3. Iterative support-and-decision process

The disparity and support values obtained in the initial support-and-decision process are used to propagate information from the visible pixels to the occluded pixels. Note that while the initial support-and-decision process relies exclusively on disparities of the visible pixels, the iterative support-and-decision process utilizes information from within the occluded region as well. The iterative support-and-decision process is similar to the initial process, except that a normalization operation is applied at each iteration. The support and normalization functions at the occluded pixels can be computed as follows.

$$\begin{aligned} S_{INT}^{t+1}(m, d) &= \sum_{n \in \mathrm{N}(m)} w(m, n) S^t(n) f(d, d(n))(1 - o(n)) \\ W(m, d) &= \sum_{n \in \mathrm{N}(m)} w(m, n) f(d, d(n))(1 - o(n)) \end{aligned} \quad (5)$$

where $S^t(n)$ represents the support function at the t^{th} iteration, and we initialize $S^0(n)$ using $S(n)$. $S_{INT}^{t+1}(m, d)$ is the intermediate (not normalized) support function, which is similar to the initial support function in Eq. (3). The occluded pixels vote for the disparity candidate of the occluded pixels by using support values to represent the reliability of the occluded pixels. In other words, since occluded pixels that have a larger support value are more likely to have reliable disparity information, they carry higher weight in determining the disparity and support values in the iterative stage. Therefore, the disparity and the support functions are computed as follows.

$$
\begin{aligned}
d^{t+1}(m) &= \arg\max_{d} S_{INT}^{t+1}(m,d) \\
S^{t+1}(m) &= S_{INT}^{t+1}(m, d^{t+1}(m))/W(m, d^{t+1}(m))
\end{aligned}
\tag{6}
$$

The final support $S^{t+1}(m)$ at the $(t+1)^{th}$ iteration is normalized by W. The normalization process is needed to preserve the sum of support values while the iterative support-and-decision process is in progress. The total sum of support values that are computed in the initial support-and-decision process should be maintained in the iterative filtering process, similar to the edge-preserving filtering methods such as bilateral filtering and non-local means filtering.

2.4. Complexity analysis

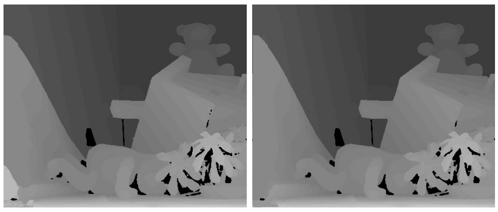
In order to evaluate the computational complexity of the proposed method, we compare it with the sequential occlusion handling method in [6]. The complexity of the sequential occlusion handling method is $O(L \cdot W \cdot D)$, where L is the number of the occluded pixels in an image, and W and D are the window size of the nonlinear filter and search range, respectively. The computational complexity depends on the search range since the method does the sequential nonlinear filtering on each cost domain repeatedly. On the contrary, the complexity of the proposed method is $O(L \cdot W)$ and is independent of the search range. Moreover, the occlusion handling process can be done hierarchically as described in the following section, which effectively enables the proposed method to operate with a relatively smaller window. The processing time of both methods will be described in the experimental results.

3. ACCELERATION TECHNIQUES

The proposed method can be accelerated with numerical methods including Gauss-Seidel acceleration and through a hierarchical scheme. For Gauss-Seidel acceleration, the updated supports in each occluded pixel are used immediately after they are computed in the iterative support-and-decision process. This makes the iterative scheme more efficient [6]. We divide a set of neighbor pixels $N(m)$ into the causal part $N_c(m)$ and the noncausal part $N_n(m)$. Using these two parts of $N(m)$, the support function in Eq. (5) can be described as follows.

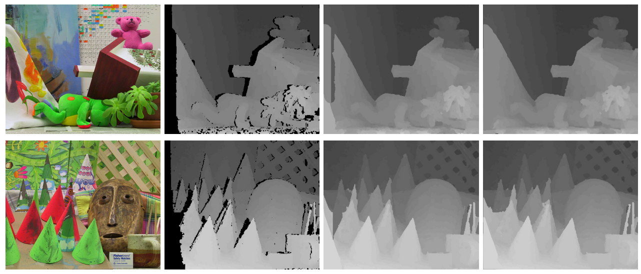
$$
\begin{aligned}
&S_{INT}^{t+1}(m,d) = \\
&\sum_{n \in \mathrm{N}_c(m)} w(m,n) S^{t+1}(n) f(d, d(n))(1 - o(n)) + \\
&\sum_{n \in \mathrm{N}_n(m)} w(m,n) S^{t}(n) f(d, d(n))(1 - o(n))
\end{aligned}
\tag{7}
$$

Another means to accelerate the proposed scheme is to apply the method hierarchically. Since it is sometimes necessary to gather support information at a large distance to ensure reliable occlusion handling, the hierarchical scheme can efficiently propagate support information between occluded pixels. In this scheme, the value that is close to the optimal support at each level is initialized by using the final value in the coarser level. In the first phase, the support function at the coarsest level is initialized using $S(m)$ in Eq. (4); the stereo images are also subsampled in order to compute weighting function w at each level. The iterative support-and-decision process is then performed at each level. After T iterations, the resulting disparity and support functions are used to initialize the disparity and support functions in the finer level. Note that in this scheme the upsampling step for the disparity and support functions is not performed since it is impossible to perform the upsampling step with the support function of occluded pixels having different disparity values.



(a) Sequential handling (b) Proposed method

Fig. 4. Occlusion handling for ground truth disparity map with occlusion in 'Teddy': Processing times are (a) $3.85s$ and (b) $0.41s$.



(a) (b) (c) (d)

Fig. 5. Results of occlusion handling for 'Teddy' and 'Cone' images: (a) left image, (b) occlusion, (c) sequential occlusion handling (d) proposed occlusion handling

4. EXPERIMENTAL RESULTS

We evaluated the performance of the proposed method and compared it with other methods using the Middlebury test set [7, 8] and stereo video sequences 'Heidelberg' (1280×720) and 'Rhinevalley' (720×576), which are available online [9]. The stereo video sequences 'Heidelberg' and 'Rhinevalley' were rectified using [10] and the disparity search range was calculated using a temporally-consistent range estimation method [11].

The proposed method is tested using the same parameters for all test images, except the window size for computing an initial support in Eq. (3). The weighting parameters are $\sigma_s = 12.0$ and $\sigma_I = 7.0$. In the iterative process, the hierarchical scheme is set to two levels and the number of iterations is 2 for all scales. The window sizes for computing initial support are 11×11 for the Middlebury images, and 23×23 for the 'Heidelberg' and 'Rhinevalley' images, respectively. The window sizes in the iterative step are 7×7 in the hierarchical scheme (for all scales), and 11×11 in the non-hierarchical scheme.

Fig. 4 shows the results of the occlusion handling for the 'Teddy' stereo image. The sequential occlusion handling [6] and proposed methods were applied to a ground truth disparity map with occlusion in order to compare the performance of occlusion handling. They yielded similar results, but the processing time of the sequential occlusion handling is nearly 9 times more than that of the proposed method.

Fig. 5 and Table 1 show the results and objective evaluation of the occlusion handling for the disparity map estimated using hierarchical belief propagation [12]. Due to lack of space, only the results of 'Teddy' and 'Cone' are shown. The occluded pixels were estimated using the cross-checking method with left and right disparity maps. The proposed method is tested with both the non-hierarchical and hierarchical schemes. Similar results for both the sequential occlusion handling and the proposed methods are shown, however, as shown in Table 2 the processing times of the sequential occlusion handling is several times of that of the proposed method, especially

Table 1. Error rate (%) for all pixels in Middlebury test images

Images	Tsukuba	Venus	Teddy	Cone
Before Occ. Han.	4.42	2.59	14.8	13.9
Seq. Handling	2.61	1.73	14.2	7.96
Proposed (Non-Hier)	2.55	1.72	13.4	10.3
Proposed (Hier)	2.50	1.66	13.1	8.13

Table 2. Processing times (s) in Middlebury test images

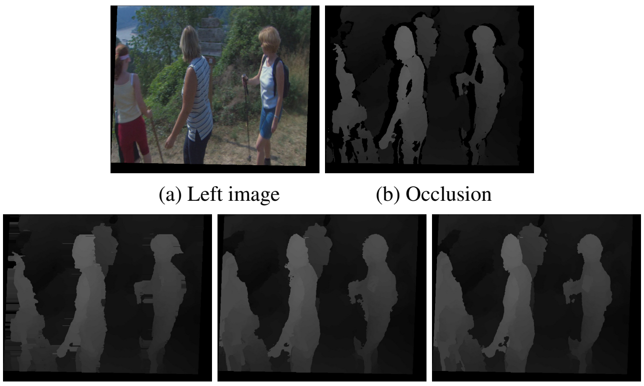
Images	Tsukuba	Venus	Teddy	Cone
Seq. Handling	$0.35s$	$0.65s$	$3.7s$	$3.9s$
Proposed (Non-Hier)	$0.22s$	$0.27s$	$0.4s$	$0.42s$
Proposed (Hier)	$0.25s$	$0.31s$	$0.38s$	$0.41s$

in the 'Teddy' and 'Cone' stereo images whose search range is large $(0, 60)$. The reason for this is that the sequential handling method does not scale well to large search ranges since it performs nonlinear filtering for all the disparity candidates repeatedly, as discussed in Section 2.4. The comparison between 'Non-Hier' and 'Hier' results show only minor improvements with the hierarchical scheme. While the hierarchical scheme can use a reduced search range, there are some additional operations such as image resampling and iterative filtering at the coarser levels that must be accounted for. It may be possible to optimize these processes further to achieve the full benefits of the hierarchical scheme.

Figs. 6 and 7 show the occlusion handling results for 'Rhinevalley' and 'Heidelberg', respectively. The estimated disparity search ranges are $(-10, 38)$ for 'Rhinevalley' and $(-4, 52)$ for 'Heidelberg' [11], and the disparity maps were estimated using the simplified method of [6]. We found that occluded pixels were generated on both sides of the foreground objects due to errors in the stereo matching methods. As shown in Figs. 6(c) and 7(b), such errors cause problems in occlusion handling schemes based on extrapolation since the processing order is based on the geometry order. The results of the proposed method were based on the hierarchical scheme and demonstrate clear improvements in accuracy compared to the sequential occlusion handling method.

5. CONCLUSION

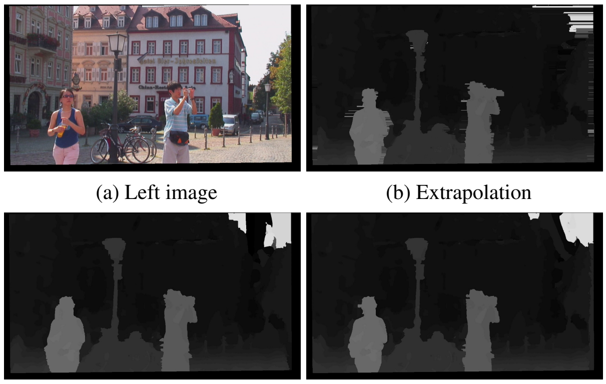
In this paper, we proposed a novel method for handling occlusion using a support-and-decision framework. The proposed method assigns disparity values to the occluded pixels using support function that represents the likelihood of having a specific disparity value. Acceleration techniques are also proposed to propagate the support information between occluded pixels efficiently. The experimental results shows that the proposed method works well for several stereo images including Middlebury test data, and its computational complexity is independent on the search range. In further research, we plan to investigate the robustness of the occlusion handling algorithm to errors in the occlusion detection. A method for temporally-consistent occlusion handling may also be considered.



(a) Left image (b) Occlusion

(c) Extrapolation (d) Sequential handling (e) Proposed method

Fig. 6. Results of occlusion handling for 'Rhinevalley' image: Processing times are (c) $0.15s$, (d) $33.8s$ and (e) $1.7s$.



(a) Left image (b) Extrapolation

(c) Sequential handling (d) Proposed method

Fig. 7. Results of occlusion handling for 'Heidelberg' image: Processing times are (b) $0.34s$, (c) $44.7s$ and (d) $3.05s$.

6. REFERENCES

[1] G. Egnal and R. P. Wildes, "Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches," *IEEE Trans. on PAMI*, 24(8), pp. 1127-1133, 2002.

[2] A.F. Bobick and S.S. Intille, "Large occlusion stereo," *International Journal of Computer Vision*, vol. 33, pp. 181-200, 1999.

[3] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 508-515.

[4] J. Sun, Y. Li, S. Kang, and H. Shum, "Symmetric stereo matching for occlusion handling," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. 399-406.

[5] L. Wang, H. Jin, R. Yang, M. Gong, "Stereoscopic Inpainting: Joint Color and Depth Completion from Stereo Images," *IEEE Proc. CVPR*, 2008.

[6] D. Min and K. Sohn, "Cost aggregation and occlusion handling with WLS in stereo matching," *IEEE Trans. Image Processing*, vol. 17, no. 8, pp. 1431-1442, Aug. 2008.

[7] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7-42, Apr. 2002.

[8] http://vision.middlebury.edu/stereo

[9] http://www.3dtv.at

[10] R. Hartley, "Theory and Practice of Projective Rectification," *IJCV*, 35(2), pp. 115-127, 1999.

[11] D. Min, S. Yea, Z. Arican and A. Vetro, "Disparity search range estimation: enforcing temporal consistency," *Proc. IEEE ICASSP*, 2010.

[12] P. F. Felzenszwalb, D. P. Huttenlocher, "Efficient belief propagation for early vision," *IJCV*, 70(1), pp.41-54 2006.