Accelerating Cost Volume Filtering Using Salient Subvolumes and Robust Occlusion Handling

Mohamed A. Helala, Faisal Z. Qureshi

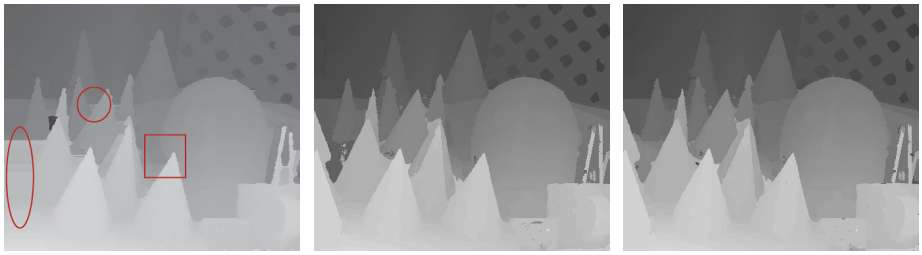
Faculty of Science, University of Ontario Institute of Technology
Oshawa, ON, Canada
{Mohamed.Helala, Faisal.Qureshi}@uoit.ca

Abstract. Several fundamental computer vision problems, such as depth estimation from stereo, optical flow computation, etc., can be formulated as a discrete pixel labeling problem. Traditional Markov Random Fields (MRF) based solutions to these problems are computationally expensive. Cost Volume Filtering (CF) presents a compelling alternative. Still these methods must filter the entire cost volume to arrive at a solution. In this paper, we propose a new CF method for depth estimation by stereo. First, we propose the Accelerated Cost Volume Filtering (ACF) method which identifies salient subvolumes in the cost volume. Filtering is restricted to these subvolumes, resulting in significant performance gains. The proposed method does not consider the entire cost volume and results in a marginal increase in unlabeled (occluded) pixels. We address this by developing an Occlusion Handling (OH) technique, which uses superpixels and performs label propagation via a simulated annealing inspired method. We evaluate the proposed method (ACF+OH) on the Middlebury stereo benchmark and on high resolution images from Middlebury 2005/2006 stereo datasets, and our method achieves state-of-the-art results. Our occlusion handling method, when used as a post-processing step, also significantly improves the accuracy of two recent cost volume filtering methods.

1 Introduction

Computing dense depth maps from a pair of stereo images is one of the fundamental problems in computer vision. In the last few years, several pixel-labeling techniques have been proposed to estimate depth maps from a pair of stereo images [3]. The goal here is to assign a depth value (or label) to each pixel, given a pair of stereo images. Pixel labeling problems are typically cast within an optimization framework, where the cost is defined for assigning a label $l \in L$ to a pixel $p \in P$. The solution to the labeling assignment problem $f : L \rightarrow P$ is then found by minimizing the overall assignment cost. Often times the desired solution is (i) spatially smooth, (ii) obeys label costs and (iii) preserves the discontinuities at image edges. Markov Random Fields (MRFs) provide a robust framework for modeling such labeling problems [5]. Overall assignment cost is modeled as an energy function that accounts for both assigning a label to a particular pixel and assigning a label pair to a pair of neighboring pixels. The pairwise term

(a) Ground Truth (b) CLMF. 7.82% (c) CF. 8.24%

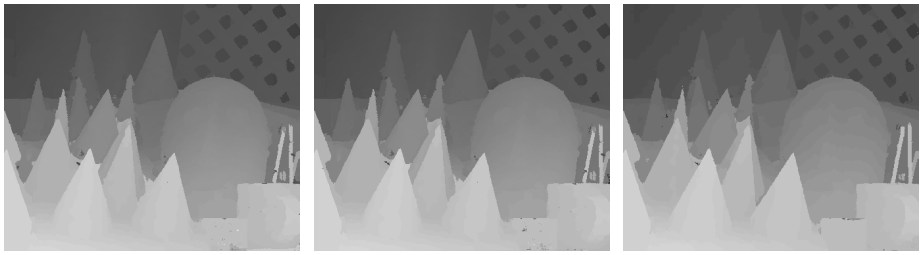


(d) VARMSOH. 9.91% (e) Ours($r = 0.2$). 7.79% (f) Ours($r = 0.3$). 7.43%

Fig. 1: Comparison on Middlebury Cones dataset [1]: (a) Ground truth, (b) CLMF [2], (c) CF [3], (d) VARMSOH [4], and (e,f) our method ACF+OH. r determines the size of local windows during salient subvolume detection. For $r = 2$, ACF+OH achieves 2.2 speedup over CF, and for $r = 3$, ACF+OH achieves 1.7 times speedup over CF. Percentage errors shown next to each figure are calculated using the default error threshold of 1.0. Ellipses and squares in CLMF, CF and VARMSOH indicate regions that exhibit large errors.

enforces spatially smooth, edge-aligned solutions. Standard energy minimization inference algorithms, such as graph cut [6] and belief propagation [7–9] yield acceptable results; however, these schemes are computationally expensive and do not fare well when dealing with large label sets or high-resolution images.

Recently local filtering methods [2, 3, 10] have been developed as an alternative to energy-based approaches. These filtering methods are designed to achieve (spatially) locally smooth label assignments, as opposed to globally smooth label assignments in the case of MRFs. Despite this simplifying assumption, recent work shows that filtering methods are able to achieve high-quality results. A benefit of these filtering methods is that their complexity is linear in the number of labels for each pixel. An early application of filtering methods for stereo appeared in [11] and [12]. Method presented in [11] was slow and did not offer any real advantage over energy-minimization methods. [12], on the other hand, employed an approximate (and fast) implementation of the filter, which resulted in a considerable speed gain at the cost of accuracy. Hosni *et al.* [3] used edge-aware *guided filters* to achieve high-quality results for multi-label problems, including stereo. Here the complexity is independent of the size of the filter, resulting in



(a) ACF+OH. 7.43% (b) CF+OH. 7.41% (c) CLMF+OH. 7.22%

Fig. 2: The proposed OH, when used in place of the RF gap-filling [3] post-processing step, improves improves the performance of ACF, CF, and CLMF methods. When using OH, instead of RF, ACF($r = 0.3$) all pixel percentage error reduces from 8.49% to 7.43%. Similarly, all pixel percentage error for CF reduces from 8.24% to 7.41%, and that of CLMF reduces from 7.82% to 7.22%. These errors are calculated using error threshold equal to 1.0. r controls the size of local windows during salient subvolume detection.

good runtime performance. Building upon this idea, Lu *et al.* [2] further improved the runtime performance by formalizing the filtering process as a local multi-point regression problem. A latter work by the same authors combined *PatchMatch* with edge-aware filtering in order to further speed up the inference process [10]. The complexity of this method is sublinear in the number of labels.

Filtering methods mentioned above rely upon a post-processing step to deal with gaps present in the initial solution.[1] These gaps exist in mismatched or non-overlapping areas. Energy minimization schemes explicitly model these gaps; whereas, filtering methods refine the initial label assignments and fill these gaps using a row-filling strategy described in [3]. These gaps seem to play a larger role in filtering methods, perhaps because these methods ignore global smoothing. This suggests that one way to increase the accuracy of filtering methods is to implement a better algorithm for gap filling. In the case of stereo, gap filling is typically referred to as occlusion handling.

Within this backdrop, this paper develops a new method for dense stereo estimation. First, we present an extension of the Cost Filter (CF) [3] method, called Accelerated Cost Filtering (ACF) (see Fig. 1). ACF uses feature matching to identify *salient subvolumes* within the cost volume and restrict filtering to these subvolumes. For stereo pairs with large disparity, this results in a significant speed up. ACF runtime performance, for example, provides a speedup of up to 4 times over CF on five high resolution images from the Middlebury 2005/2006 stereo datasets [13]. Since ACF restricts filtering to the selected subvolumes, initial label assignments show a marginal increase in the gaps as compared to those returned from CF. We develop an Occlusion Handling (OH) (or gap filling) method that uses superpixels [14] and a label propagation algorithm inspired by

[1] Gap here refers to pixels with no label assignments.

simulated annealing [15] to propagate the labels to pixels within the occluded regions (see Fig. 2). Our occlusion handling method gives better results than the row filling method described in [3].

The following hypotheses underpin our work on ACF: 1) each slice in the cost volume can be partitioned into visible and non-visible regions, where the visible regions indicate areas where input images agree (*visibility hypothesis*) and 2) matched keypoints in the stereo image pair can be used to define the visible regions within a slice of the cost volume (*selection hypothesis*). Edge aware filtering methods implicitly assume that the visibility hypothesis stands. Good feature point matches between the two images is needed for the second hypothesis to hold. For this work, we use Lowe's Scale Invariant Feature Transform (SIFT) keypoints [16, 17] to find matches between the two images. Given a matched keypoint between the left and right images, it is possible to compute the disparity for its location and use it to define a salient subvolume within the cost volume.

Our algorithm for occlusion handling uses superpixels. Superpixels are robust to noise, respect object boundaries and encode within them a higher-level of image representation. We use the SLIC algorithm that was proposed in [18] to generate superpixels. Label propagation to pixels in the occluded regions is modeled as *simulated annealing*, where appearance similarity between neighboring superpixels determine the temperature. Initially temperature is high and labels are propagated to most similar neighboring superpixels. However, by decreasing the temperature, it is possible to propagate labels to superpixels with lower similarity values. We show that this has the advantage of assigning consistent labels that preserve edge discontinuities in the resulting disparity maps.

We have compared our ACF+OH method with the Cost Filter (CF) [3], Cross-based Local Multipoint Filtering (CLMF) [2], and a recent global energy minimization method (VARMSOH) that appeared in [4] on the Middlebury stereo benchmark dataset [1]. We also compare ACF+OH with CF [3] on five high resolution images (Rocks1, Rocks2, Dolls, Moebius, and Books) from Middlebury 2005/2006 stereo datasets [13], and our method achieves state-of-the-art results. The results also demonstrate that our occlusion handling method improves the accuracy of CF on all error measures and that of CLMF on the error percentage of all and non-occluded image regions. We have not compared our method with [10], which uses slanted surfaces.

1.1 Contributions and Outline

Our contributions are threefold. First, we develop an algorithm for computing salient subvolumes within the cost volume for label assignment problems via filtering. Second, we present a gap filling (occlusion handling) algorithm that gives better results than existing gap filling strategy proposed in [3]. We show that our gap filling algorithm can be used as a post-processing step to refine the initial label assignments in other filtering methods. We used our gap filling technique to refine the label assignments returned by CF [3] and CLMF [2] and show that our method is superior to the row-filling method for occlusion handling. Third, we extend the CF method incorporating ideas developed in

this work and show that our algorithm achieves state-of-the-art results on the Middlebury benchmark dataset [1], beating [2], [3] and [4]. We also beat [3] on high resolution images from the Middlebury 2005/2006 stereo datasets [13].

The rest of the paper is organized as follows. We discuss related work in the next section. The following section describes the methodology. We present experimental results in Sec. 4 and concludes the paper with a summary and discussion in the following section.

2 Literature Review

MRF global energy minimization techniques [4, 5, 19] are popular approaches for solving pixel labeling problems. These approaches, however, do not scale well with large cost volumes. Edge-aware filtering methods [20, 21] have contributed to the development of fast alternative techniques for solving pixel labeling problems [3, 2]. These technique are broadly referred to as cost volume filtering methods. Hosni *et al.* [3], for example, provide a framework that uses a guided filter [20] for cost volume filtering. The complexity of their approach is independent of the size of the filter. Lu *et al.* [2] further speed up the filtering process by aggregating cost estimates for a set of points. Cost volume filtering methods—although more efficient than MRF global energy minimization schemes—scale linearly with the size of the cost volume, which is undesirable when dealing with large cost volumes.

There have been several attempts to reduce the complexity of cost volume filtering. For example, Min *et al.* [22] proposed the histogram-based disparity pre-filtering scheme that reduces the cost aggregation by estimating the set of most likely candidate disparities for each pixel. This method, however, requires a pre-scanning of the entire cost volume. Lu *et al.* [10] proposed a method that combines EAF with the randomized search of *PatchMatch* to speed up the filtering process. This method has sublinear complexity in the label space size. Boufama *et al.* [23] developed a fast method for dense matching, which can perhaps be used for disparity map calculation.

A number of occlusion handling techniques have been proposed for gap filling. For example, Sun *et al.* [24] formulated stereo matching as a global energy minimization problem and added an extra term to enforce smoothness of occlusions. Min *et al.* [19] proposed an energy minimization method that filled occluded regions through label propagation. Yang *et al.* [25] formulated an energy minimization problem that used an iterative refinement step to fit planes to color segmented regions. It then filled occluded regions by minimizing the difference to the fitted planes. The work of Ben-Ari *et al.* [4] also provided an energy minimization formulation that explicitly model occlusions using an energy term, which was optimized by an iterative scheme. Hosni *et al.* [3] proposed a post-processing method for occlusion handling. This method performs row scanning and assigns each occluded pixel the lowest disparity value among its neighbouring non-occluded pixels. Weighted median filtering is employed to remove undesirable artifacts in the resulting disparity map.

3 Methodology

We begin by discussing the cost volume filtering method for estimating disparity $D(x, y)$ from stereo image pair (I_1, I_2). It is straightforward to estimate the depth of a pixel given a disparity map. Without the loss of generality we treat I_1 as the reference image I_{ref} and disparity map $D(x, y)$ assigns a disparity value to every pixel (x, y) in the reference image. We will also denote I_2 as I_k to keep open the possibility of using more than two images for estimating the disparity maps.

3.1 Cost Volume Filtering

Depth estimation given a stereo image pair can be reformulated as a discrete label assignment problem, where each pixel p with coordinates (x, y) is assigned a label l_p. Here $l_p \in L$ and L is the set of pixel disparities. Disparity space image, often referred to as cost volume, $C(x, y, l)$ is defined over pixels and the set of possible labels [1, 3]. Cost volume stores the costs of assigning a label l to a pixel at (x, y). Each slice of the cost volume is filtered, i.e., the cost of assigning a label l to a pixel at (x, y) is the weighted average of the costs of assigning label l to the neighboring pixels. Mathematically,

$$C'(x, y, l) = \sum_{(u,v)\in[-\frac{w}{2},\frac{w}{2}]} W_{I_{\text{ref}}(x,y)}(u, v)C(x + u, y + v, l), \tag{1}$$

where w is the size of the kernel, $W_{I_{\text{ref}}(x,y)}(u, v)$ are weights determined using the guidance image for each pixel location (x, y), and $C'(x, y, l)$ represent the filtered costs of assigning label l to the pixel at location (x, y). In the case of stereo, the guidance image is the reference image. Various filtering techniques exist to select $W_{I_{\text{ref}}(x,y)}(u, v)$ while preserving the intensity changes of the guidance image during the filtering process [20, 21]. Finally, a *winner-takes-all* scheme is applied to assign a label l_p to a pixel p at (x, y). Specifically,

$$l_p = \arg\min_l C'(x, y, l). \tag{2}$$

Aside: Although edge aware filtering techniques can provide accurate results in a fast and efficient manner, they need to process the entire cost volume. Therefore, the complexity of edge aware filtering is linear in the number of cost volume slices (or labels). [10] attempted to remedy this situation and combined edge aware filtering with randomized search; however, their runtime performance is similar to that of [2] on the Middlebury dataset. An obvious strategy to improve runtime performance is to restrict filtering to small sections of the cost volume.

3.2 Salient Regions in the Cost Volume

We now describe our algorithm for selecting salient subvolumes within a cost volume. Filtering is restricted to these subvolumes, resulting in reduced processing

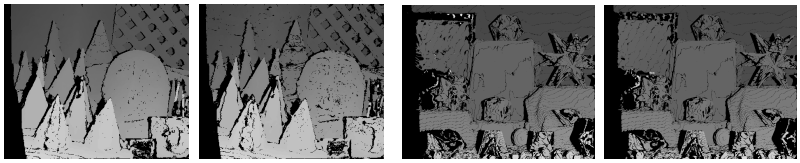
Fig. 3: Using feature matching to identify salient regions in the cost volume. The first two images from left show projections at disparities 10.8 and 14.4, respectively. For each image, we show how plane keypoints (shown as blue) indicate best matching locations for each depth. The next two images show local windows (red squares) around each plane keypoint and the final salient region (black rectangle) for these planes. We also show the expanded windows from neighboring planes as white rectangles centered on white keypoints from these planes. The width of local windows around each keypoint is equal to $r \times I_{\text{width}}$. For this figure r is set to .2.

times and better runtime performance. Our method constructs the cost volume $C(x, y, l)$ using the fronto-parallel plane sweep algorithm from [26]. The family of depth planes is defined within the coordinate system of the reference image I_{ref}. The depths l of the planes fall within the expected disparity range. For the *Teddy* dataset, 187 equally spaced fronto-parallel planes were constructed with depths ranging between 1 and 57. In order to compute $C(x, y, l)$, the pixel (x, y) in the reference image I_{ref} is projected to the other image(s) I_k using homography that relates I_{ref} to I_k via the fronto-parallel plane at depth l (for details, please see [26]). Similar to [3], if pixel (x, y) is mapped to location (x_k, y_k) in image I_k then

$$\begin{aligned} C(x, y, l) = (1 - \beta) \min(\|I_{\text{ref}}(x, y) - I_k(x_k, y_k)\|, \gamma_1) \\ + \beta \min(\|\nabla_x I_{\text{ref}}(x, y) - \nabla_x (I_k(x_k, y_k)\|, \gamma_2). \end{aligned} \tag{3}$$

$\beta \in [0, 1]$, γ_1 and γ_2 are user-defined thresholds. The intuition behind this formulation is that when the surface projected to pixel (x, y) intersects the plane at l, $I_{\text{ref}}(x, y)$ and $I_k(x_k, y_k)$ should have similar appearances under the Lambertian surface assumption.

Salient regions are defined within the cost volume by inspecting each plane in the cost volume and identifying sections where $I_{\text{ref}}(x, y)$ agrees well with $I_k(x, y)$. One scheme to find such sections in a plane at l is to use already computed $C(x, y, l)$. A better method, however, is to employ feature matching. Our method extracts SIFT keypoints from the input images (I_{ref} and I_k) and compares these using the ratio test to find matches between the two images [16, 17]. It is easy to calculate the disparity values between matched points' pairs, identifying locations (x, y, l') within the cost volume that correspond to each matched points' pair. The salient region for the plane at depth l' is constructed as follows: 1) define local windows $b_{l'}(x, y)$ centered around each (x, y, l') and 2) construct the smallest window $b_{l'}$ that encloses all $b_{l'}(x, y)$ defined in the previous step. The dimensions of local windows $b_{l'}(x, y)$ is typically a small fraction r of



(a) CF. 17.6% (b) ACF. 20% (c) CF. 30.35% (d) ACF. 29.23%

Fig. 4: Initial disparity maps computed by our ACF method shows a slight increase in the percentage of occluded pixels; black regions are occluded pixels. (Cones): (a) and (b) show the disparity maps computed by CF and ACF($r = .3$), respectively, without any gap filling. (Moebius): (c) and (d) show high resolution disparity maps computed by CF and ACF($r = .3$), respectively, without any gap filling. It is interesting to note that for Moebius dataset, ACF creates disparity maps with less occluded pixels. One explanation might be that filtering reduced cost volume may be better for some scenes. More work is needed to investigate this behavior. The width of local windows used for constructing salient regions is equal to $r \times I_{\text{width}}$.

the dimensions of the reference image. For the experiments presented here, the widths of the local windows are roughly 0.2 to 0.3 times the image width I_{width}.

Often times the depths l' of (x, y, l') locations corresponding to the matched points' pairs do not match exactly with a depth plane available for the cost volume (remembering that depth planes are simply a discrete representation of the cost volume). Such situations are dealt with by considering (x, y, l') for more than one neighboring depth planes. Specifically, each location (x, y, l') is used during computing the salient regions b_l for planes with depths l, such that $\|l - l'\| \leq u$. u is a user-defined parameter that controls the expansion within the disparity range. Note that, the definition of salient regions does not depend on the cost volume $C(x, y, l)$. So, a better strategy is to pre-compute the salient regions and only build the cost volume for these regions.

Fig. 3 illustrates our method for identifying salient regions in the cost volume. The two images on the left show the results of projecting stereo image pair from the Cones dataset on the planes at different depths (assuming different disparities) in the cost volume. Notice that different regions of the projection appear to be in focus at different depths as expected. We also show the locations of the (matched) keypoints (as blue dots). Notice that disparity calculated for a matched keypoint agrees with the depth at which the neighboring area is in focus. The two images on right show the final salient region (black rectangle) for the two depth levels. The final salient region is determined by computing the minimum bounding box for the local windows around the keypoints whose disparity agrees with the depth level, and the expanded local windows from the neighboring depth planes.

Together the salient regions for neighboring depth levels define a cuboid within the cost volume. This process results in (ideally) a sparse set of cuboids within the cost volume, and the subsequent filtering is restricted to these sub-

volumes, which results in an increased runtime performance. A side effect of restricting filtering to these subvolumes is that resulting disparity maps show a fractional increase in the number of unlabeled pixels. Parallax effects manifest themselves as unlabeled pixels. Similarly boundary regions that are missing from one or the other image of the stereo pair also lead to gaps or unlabeled pixels. We employ the method proposed in [3] to identify the pixels with missing or incorrect disparity values. Given a stereo pair, two disparity maps are constructed. The first disparity map D_1 is constructed treating I_1 as the reference image; where as, the second disparity map D_2 is constructed using I_2 as the reference image. A pixel (x, y) is considered unlabeled if its disparity values in D_1 and D_2 do not agree. For the remaining of this discussion, we will refer to the unlabeled pixels as occluded pixels. By extension, pixels that are correctly labeled will be henceforth referred to as non-occluded pixels. Fig. 4 shows the disparity maps constructed using our method compared to the ones constructed by [3]. For our disparity maps, filtering is only performed for the selected subvolumes within the cost volume. The black regions indicate occluded pixels.

3.3 Gap Filling

We now describe our gap filling method for computing disparity values for occluded pixels. Our gap filling algorithm relies upon superpixels. The intuition behind our framework stems from the following three observations: 1) the solutions of pixel-labeling problems are spatially smooth and preserve discontinuities at image edges; 2) spatially compact superpixels preserve boundaries and increase the chance that neighboring pixels within a superpixel share similar labels (or disparity values); and 3) using superpixels as primitive units boosts the runtime performance. In this work, we use the SLIC superpixel segmentation algorithm that appeared in [18] to segment an input image I into a set $\mathcal{S} = \{S_1, S_2, S_3, \cdots, S_K\}$ of K non-overlapping superpixels. The SLIC method scales linearly with the size of the image and creates compact superpixels that respect image edges. Each superpixel is defined over a set of contiguous coordinates (x, y) and $(x, y) \in S$ denotes that pixel at (x, y) belongs to superpixel S. $(x, y) \in [1, I_{\text{width}}] \times [1, I_{\text{height}}]$, and I_{width} and I_{height} represent the width and the height of image I.

The proposed method begins by assigning occlusion probabilities $p_{\text{occ}}(S)$ to each superpixel $S \in \mathcal{S}$. These probabilities are determined using the disparity map $D(x, y)$ computed in the previous step. A superpixel is said to be non-occluded if none of its pixels are occluded. Say $Occ(x, y) = 1$ for pixels that are occluded and zero otherwise. We can then define the occlusion probability of a superpixel as follows:

$$p_{\text{occ}}(S) = \frac{\sum_{(x,y)\in S} Occ(x, y)}{|S|}, \tag{4}$$

$|S|$ denotes the number of pixels in the superpixel $S \in \mathcal{S}$. $p_{\text{occ}}(S)$ is 0 if the superpixel is not occluded. Function $h(S)$ returns the most likely label for a

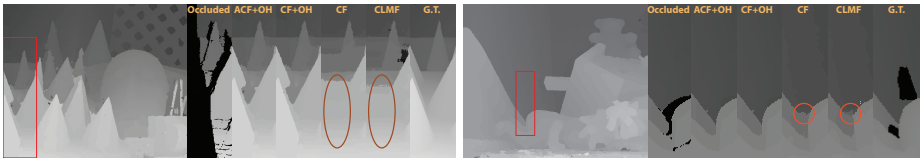


Fig. 5: Depth estimation using cost filtering with and without our OH method on the Cones (left) and Teddy (right) Middlebury benchmark datasets. For each dataset, the left-most figure shows the depth map computed by our method (ACF+OH). The other columns show a close-up of the section highlighted as the red rectangle. Occluded column shows the depth map computed by ACF without any post-processing. ACF+OH post-processes depth computed by ACF using OH. Similarly, CF+OH indicates the result of CF after post-processing using OH. CF and CLMF show depth maps computed by these algorithms, respectively. The last column shows the Ground Truth (G.T.). Ellipses in CF and CLMF indicate regions that exhibit large errors.

superpixel S when $p_{\text{occ}}(S) < 1$. $h(S)$ is simply the most frequent label $D(x, y)$ where $(x, y) \in S$. Unlabeled pixels $(x, y) \in S$ when $p_{\text{occ}}(S) < \tau_{\text{fill}}$ are set equal to $h(S)$. After this step $\forall S \in \mathcal{S}_{\text{occ}}$, $p_{\text{occ}}(S) \geq \tau_{\text{fill}}$. τ_{fill} is a user-defined threshold. In our examples, it is set to either 0.5 or 0.6. The set of superpixels can be partitioned into occluded $\mathcal{S}_{\text{occ}}$ and non-occluded $\mathcal{S}_{\text{nocc}}$ superpixel sets, such that $\mathcal{S} = \mathcal{S}_{\text{occ}} \cup \mathcal{S}_{\text{nocc}}$.

Label Propagation via Simulated Annealing. Superpixels provide a good starting point for label propagation. Pixels within the same superpixels tend to have similar labels (under the local smoothness assumption) and the superpixels align with scene intensity edges. The label propagation method defines an adjacency graph $G = (\mathcal{S}, \mathcal{S} \times \mathcal{S})$ over the set of superpixels. The graph has as its nodes, the superpixels, with edges between any two superpixels if they share a part of their boundaries. Given a superpixel S, $N_{\text{nocc}}(S)$ is the possibly empty set of its neighboring non-occluded superpixels. It is straightforward to construct $N_{\text{nocc}}(S)$ given G. Let $sim(S, S') = 1 - \|col(S) - col(S')\|_2 \in [0, 1]$ defines a similarity value between two neighboring superpixels, where $col(S)$ and $col(S')$ return the normalized average colors for superpixels S and S', respectively. The label propagation algorithm is inspired by *simulated annealing* [15]. The similarity threshold T at which labels are propagated to neighboring superpixels is slowly reduced over time by ΔT. The labels for non-occluded superpixels are never updated. The following algorithm describes our label propagation algorithm:

Require: $\mathcal{S}_{occ}, \mathcal{S}_{nocc}, T, \Delta T$
Ensure: $\mathcal{S}_{occ} = \Phi$ and $\mathcal{S} = \mathcal{S}_{nocc}$
1: $T = 1.0$;
2: **while** $\mathcal{S}_{occ} \neq \Phi$ **do**
3: **for all** $S \in \mathcal{S}_{occ}$ **do**
4: **if** $N_{\text{nocc}}(S) = \Phi$ **then**

```
      continue
    end if
    S* = arg max_{S'} sim(S, S'), where S' ∈ N_nocc(S)
    if sim(S, S*) > T then
      ∀(x, y) ∈ S, if Occ(x, y) then D(x, y) = h(S*)
      S_nocc = S_nocc ∪ {S}
      S_occ = S_occ − {S}
    end if
  end for
  T = max(T − ΔT, 0.0)
end while
```

As a final enhancement step, we apply the weighted median filtering used in [3]. Fig. 5 shows results for our occlusion handling method. Note that the results from ACF+OH method are closer to the ground truth than those of CF and CLMF. Furthermore, the proposed OH, when used as a post-processing step for CF, improves its results.

4 Results

We have compared our ACF+OH method against CF [3], CLMF [2], VARMSOH [4] methods on the Middlebury stereo benchmark dataset [1]. We also compared our method against CF [3] on the Rocks1, Rocks2, Moebius, Dolls, and Books high resolution Middlebury 2005/2006 datasets [13]. Note that, VARMSOH applies global energy minimization for occlusion handling; whereas, CF and CLMF use Row Filling (RF) [3]. In our results, we will indicate CF and CLMF, as CF+RF and CLMF+RF.

Table 1 lists quantitative stereo evaluation results on Middlebury benchmark. It shows rank and average percentage error corresponding to two error thresholds: 1 and 0.5 (default error threshold is 1.0). Notice that our method ACF+OH outranks CF+RF, CLMF+RF, and VARMSOH methods on the default threshold. For error threshold equal to 0.5, our method performs slightly worse than CF+RF and VARMSOH. This is because currently our method does not support slanted planes. We plan to address this limitation in the future. The table also shows the importance of occlusion handling for our method. Notice that ACF+RF's rank drops to 60 and 64 for $r = .2$ and $r = .3$, respectively, for the default error threshold. The table also shows that the proposed OH method improves the performance of CF algorithm—CF+OH's rank is 25 as compared to that of CF+RF, which is 42 for the default threshold. CLMF+OH and CLMF+RF achieve similar performance. Furthermore, on high-resolution Middlebury datasets, ACF+OH and CF+RF achieve average percentage of all pixel errors of 10.57% and 11.13%, respectively. These results support the central premise of this paper: it is possible to filter sub-volumes in the cost volume and achieve accuracy comparable to schemes that filter the entire cost volume.

Table 2 compares ACF and CF without any post-processing steps on the Middlebury standard and high resolution datasets. Notice that while average

Table 1: Quantitative evaluation on Middlebury benchmark datasets [1]. These results are aggregated over Cones, Teddy, Venus, and Tsukuba datasets.

Algorithm	Error thrreshold = 1		Error thrreshold = 0.5	
	Rank	% error	Rank	% error
CF+OH	**25**	5.22	30	12.9
CLMF+OH	38	5.14	66	16.9
ACF+OH ($r = .3$)	30	5.26	33	13
ACF+OH ($r = .2$)	39	5.45	37	13.3
CF+RF [3]	42	5.55	27	12.8
CLMF+RF [2]	37	**5.13**	64	16.7
ACF+RF($r = .3$)	64	5.99	45	13.4
ACF+RF($r = .2$)	60	5.92	42	13.6
VARMSOH [4]	116	8.17	**21**	**11.8**

Table 2: A comparison of ACF and CF without any post-processing step on Middlebury standard and high-resolution datasets. Runtimes for RF and OH post-processing steps are also provided.

Algorithm	Average % occluded pixels		Run-time (seconds)	
	Standard	High Resolution	Standard	High Resolution
ACF(r=0.2)	14.2	-	16.117	-
ACF(r=0.3)	14.39	26.1	18.717	159.82
CF	13.6	26.9	28.2	505
RF	-	-	0.11	1.4
OH	-	-	0.131	0.2

percentage occluded pixel values for ACF and CF are comparable, ACF's runtime performance is significantly better than that of CF's, especially on high resolution datasets. This table also lists runtimes for RF and OH post-processing steps. Our post-processing method outperforms RF on high resolution datasets.

Table 3 shows the Middlebury stereo evaluation results on the Middlebury benchmark datasets for the default error threshold 1.0. For each dataset, we list the values for the three popular error measures: 1) nocc, which measures the error percentage of non-occluded regions, 2) all, which calculates the error percentage of all regions, 3) disc, which provides the error percentage of regions near depth discontinuities. Notice that our method ACF+OH outperforms CF+RF and CLMF+RF on nearly every error measure when $r = .3$, and on the error percentage of all regions when $r = .2$. It also outperforms VARMSOH on all measures. The results also show that the proposed OH method improves ACF over the RF scheme—E.g. for the Cones dataset, ACF+OH has an all error of 7.79% and 7.43% for $r = .2$ and $r = .3$, respectively; however, for ACF+RF, these errors are 9.06% and 8.49%. Additionally OH significantly improves the accuracy of CF on every error measure, and that of CLMF on the nocc and all error measures.

Table 4 list the OH parameters used for results presented here. ΔT is set to 0.0001 in all cases. While the performance of our occlusion handling depends

Table 3: Stereo evaluation results on Middlebury benchmark with error threshold equal to 1.0.

Algorithm	Tsukuba			Venus			Teddy			Cones		
	nocc	all	disk	nocc	all	disk	nocc	all	disc	nocc	all	disk
CF+OH	**1.45**	**1.75**	7.37	**0.19**	**0.37**	2.24	5.85	**10**	16.1	2.6	7.41	7.31
CLMF+OH	2.39	2.69	6.53	0.26	**0.37**	2.23	**5.49**	10.7	**14.2**	2.46	**7.22**	7.10
ACF+OH ($r = .3$)	1.45	1.75	7.37	0.19	0.37	2.24	5.94	10.1	16.4	2.61	7.43	7.23
ACF+OH ($r = .2$)	1.45	1.75	7.37	0.19	0.37	2.24	6.64	10.7	16.3	2.82	7.79	7.74
CF+RF [3]	1.51	1.85	7.61	0.2	0.39	2.42	6.16	11.8	16	2.71	8.24	7.66
CLMF+RF [2]	2.46	2.78	**6.26**	0.27	0.38	**2.15**	5.50	10.6	**14.2**	**2.34**	7.82	**6.80**
ACF+RF ($r = .3$)	1.51	1.85	7.61	0.2	0.39	2.42	6.94	11.3	18.5	3.38	8.49	9.3
ACF+RF ($r = .2$)	1.51	1.85	7.61	0.2	0.39	2.42	6.96	11.1	17.1	3.66	9.06	9.8
VarMSOH [4]	3.97	5.23	14.9	0.28	0.76	3.78	9.34	14.3	20	4.14	9.91	11.4

Table 4: Parameters for OH procedure. $\Delta T = 0.0001$.

Dataset	#superpixels	τ_{fill}	Dataset	#superpixels	τ_{fill}	Dataset	#superpixels	τ_{fill}
Cones	1600	0.6	Teddy	2000	0.6	Tsukuba	500	0.5
Venus	1000	0.5	Rocks1	700	0.5	Rocks2	700	0.5
Moebius	1600	0.5	Dolls	1600	0.5	Books	1600	0.5

upon these parameters, plots in Fig. 6 suggest that the proposed OH method is able to achieve good accuracy over a range of these parameters. Expansion factor u is chosen to be 2 and 6, respectively, for the standard and high-resolution datasets. Figure 6 plots the accuracy of CF+RF and CF+OH methods against the τ_{fill} user-selected threshold (for OH method) for the Cones (top-row) and Teddy (bottom-row) datasets. Accuracies are plotted for different values of the number of superpixels K. These plots suggest that the proposed OH method improves both all and nocc errors, over a range of values for τ_{fill} and K.

Figure 7 shows a run-time vs. accuracy comparison for our ACF+OH method while varying the size of local windows used for defining salient regions. The local window size is expressed as a fraction $r \times I_{\text{width}}$ (along x-axis). The y-axis represents run-times in seconds. As expected the accuracy increases when using large window sizes for computing salient regions. The good news is that the accuracy does not change much when using window sizes that are more than $0.3 \times I_{\text{width}}$.

5 Conclusions

This paper develops accelerated cost volume filtering for disparity estimation from a stereo image pair. Feature matching is used to identify salient subvolumes within the cost volume and filtering is restricted to these subvolumes, resulting in increased runtime performance. We have also developed an occlusion handling method, which acts as a post-processing step and refines the disparity maps computed via filtering. The occlusion handling technique relies upon superpixels

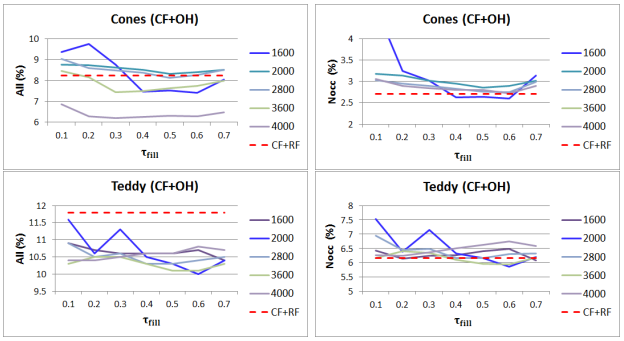


Fig. 6: The accuracy of CF+RF and CF+OH against τ_{fill} threshold for OH method. The dashed (red) line indicate the accuracy of CF+RF. It is independent of the choice of τ_{fill}. Solid lines indicate the all (left column) and nocc (right column) percentage errors for different values of the number of superpixels K. The top row shows plots for Cones dataset and the bottom row shows plots for Teddy dataset. *This figure is best viewed in color.*

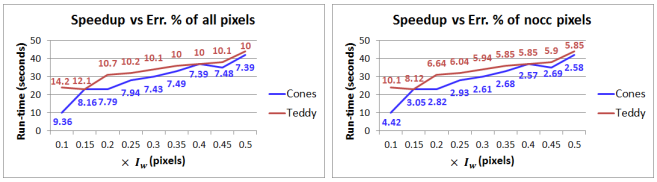


Fig. 7: Run-time vs. accuracy comparison for our ACF+OH method using different values for the r parameter that controls the size of local window used for defining salient regions. The numbers printed next to the plots represent average percentage errors. *This figure is best viewed in color.*

and uses a simulated annealing inspired method for label propagation between superpixels, preserving edge discontinuities in the process. The proposed method is evaluated on the Middlebury stereo datasets and it outperforms state-of-the-art techniques: CF [3], CLMF [2] and VARMSOH [4]. Our occlusion handling method also improves the accuracy of CF on all error measures and that of CLMF on the error percentage of all and non-occluded image regions. In the future we intend to explore the use of slanted surfaces during detecting salient subvolumes. We also hope to apply our method to other discrete labeling problems, such as optical flow computation, etc.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision **47** (2002) 7–42
2. Lu, J., Shi, K., Min, D., Lin, L., Do, M.: Cross-based local multipoint filtering. In: Proc. IEEE CVPR. (2012) 430–437

3. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2013) 504–511
4. Ben-Ari, R., Sochen, N.: Stereo matching with mumford-shah regularization and occlusion handling. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 2071–2084
5. Delong, A., Osokin, A., Isack, H., Boykov, Y.: Fast approximate energy minimization with label costs. International Journal of Computer Vision **96** (2012) 1–27
6. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 1222–1239
7. Weiss, Y., Freeman, W.: On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. IEEE Transactions on Information Theory **47** (2001) 723–735
8. Sun, J., Zheng, N., Shum., H.: Stereo matching using belief propagation. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 787–800
9. Felzenszwalb, P., Huttenlocher, D.: Efficient belief propagation for early vision. International Journal of Computer Vision **70** (2006) 41–54
10. Lu, J., Yang, H., Min, D., Do, M.: Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In: Proc. IEEE CVPR. (2013) 1854–1861
11. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Transactions on Pattern Analysis and Machine Intelligence **28** (2006) 650–656
12. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.: Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In: ECCV 2010. Volume 6313 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 510–523
13. Hirschmuller, H., Scharstein, D.: Evaluation of cost functions for stereo matching. In: Proc. IEEE CVPR. (2007) 1–8
14. Schick, A., Bauml, M., Stiefelhagen, R.: Improving foreground segmentations with probabilistic superpixel markov random fields. In: Proc. IEEE CVPRW. (2012) 27–31
15. Granville, V., Krivanek, M., Rasson, J.: Simulated annealing: a proof of convergence. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 652–656
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
17. Brown, M., Hua, G., Winder, S.: Discriminative learning of local image descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011) 43–57
18. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Susstrunk, S.: SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 2274–2282
19. Min, D., Sohn, K.: Cost aggregation and occlusion handling with WLS in stereo matching. IEEE Transactions on Image Processing **17** (2008) 1431–1442
20. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 1397–1409
21. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral filtering: Theory and applications. Foundations and Trends in Computer Graphics and Vision **4** (2009) 1–73
22. Min, D., Lu, J., Do, M.: A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy? In: Proc. IEEE ICCV. (2011) 1567–1574

23. Boufama, B., Jin, K.: Towards a fast and reliable dense matching algorithm. Society of Manufacturing Engineers Journal (2003)
24. Sun, J., Li, Y., Kang, S., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: Proc. IEEE CVPR. Volume 2. (2005) 399–406
25. Yang, Q., Wang, L., Yang, R., Stewenius, H., Nister, D.: Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (2009) 492–504
26. Gallup, D., Frahm, J.M., Mordohai, P., Qingxiong, Y., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions. In: Proc. IEEE CVPR. (2007) 1–8