Improved Color Patch Similarity Measure Based Weighted Median Filter

Zhigang Tu, Coert Van Gemeren, and Remco C. Veltkamp(✉)

Department of Information and Computing Sciences, Utrecht University,
Utrecht, The Netherlands
{Z.Tu,C.J.Vangemeren,R.C.Veltkamp}@uu.nl

Abstract. Median filtering the intermediate flow fields during optimization has been demonstrated to be very useful for improving the estimation accuracy. By formulating the median filtering heuristic as non-local term in the objective function, and modifying the new term to include flow and image information that according to spatial distance, color similarity as well as the occlusion state, a weighted non-local term (a practical weighted median filter) reduces errors that are produced by median filtering and better preserves motion details. However, the color similarity measure, which is the most powerful cue, can be easily perturbed by noisy pixels. To increase robustness of the weighted median filter to noise, we introduce the idea of non-local patch denoising method to compute the color similarity in terms of patch difference. Most importantly, we propose an improved color patch similarity measure (ICPSM) to modify the traditional patch manner based measure from three aspects. Comparative experimental results on different optical flow benchmarks show that our method can denoise the flow field more effectively and outperforms the state-of-the art methods, especially for heavy noise sequences.

1 Introduction

Estimation of a dense motion field between video frames plays a fundamental role in computer vision and image processing. One of the most successful techniques that address this problem is the variational optical flow method [2,3,9], and it has been widely used for various visual tasks, such as tracking, object segmentation and recognition, and super-resolution reconstruction. Since the seminal work of Horn-Schunck (HS) [1], various subsequent extensions and improvements have been proposed over the past 30 years to tackle drawbacks of the HS model, and there has been tremendous progresses, such as: pre-processing the input images with photometric invariant constraints or structure-texture decomposition technique to handle illumination changes [11], or with pre-filtering approaches (e.g. Gaussian filter [1] and Laplacian filter [3]) to reduce outliers (e.g., image noise and estimated flow errors). Penalty functions [2,4,5] are utilized to preserve motion discontinuities and increase robustness to outliers and occlusions. Additionally, occlusions can also be handled according to bilateral filters [6,7]. Large displacements can be estimated by employing traditional coarse-to-fine

D. Cremers et al. (Eds.): ACCV 2014, Part V, LNCS 9007, pp. 413–427, 2015.
DOI: 10.1007/978-3-319-16814-2_27

strategy [8] or matching techniques (e.g. sparse feature matching [9] and dense correspondence matching [10]). However, there are still outstanding problems in existing optical flow methods, such as outliers and large displacements. This paper addresses the issue of outliers removing in terms of median filtering.

In a current survey paper, in comparing various modern optimization and implementation techniques, Sun *et al.* [5] point out that applying a median filter [11,12] to intermediate flow values during incremental estimation and warping produces the most significant improvements. Median filtering is beneficial for every optical flow algorithm they tested, since it can effectively denoises the intermediate flow fields and reduces gross outliers. All in all, it can make even non-robust methods much more robust.

However, median filtering in a large neighborhood has negative effects on edges and corners. A neighborhood centered on a corner or thin structure is dominated by its surroundings, leading to oversmoothing. To improve the performance of the classic median filtering [5,11,12] formulated the median filtering heuristic as a non-local term in the objective function, and incorporated flow and image information to construct a weighted version (a practical weighted median filter). The weighted non-local term is very useful to avoid smoothing at edges and motion boundaries, and can well preserve motion details. Because pixels that belong to the same surface are given higher weight and it ensures pixels only propagate information within their same region.

Although the weighted median filter (WMF) has great advantages, it still has one serious problem. As the weight of WMF heavily depends on the color similarity between pixels, it can be easily violated by noisy pixels [13]. Consequently, improving the robustness of the color similarity measure to noise is a good way to improve the performance of the WMF.

Since Buades *et al.* [14] proposed a non-local means (NLM) method, which uses patches instead of pixels to compare photometric similarities, the non-local denoising methods [15,16] have attracted a lot of attention recently and outperforms conventional filters, leading to the patch-based non-local manner becoming the central part of many state-of-the-art algorithms [17,18]. The NLM denoises a pixel as the weighted sum of its noisy neighbors, where each weight reflects the similarity between the local patch centered at the noisy pixel (i, j) to be denoised and the patch centered at the neighbor pixel (i', j') [15]. In this way, the NLM not only compares the intensity in a single point but also the geometrical configuration in a whole neighborhood. This characteristic allows a more robust comparison than traditional neighborhood filters, and pixels with a similar intensity neighborhood to the noisy pixel will assign higher weights on average.

In this paper, we adopt the idea of the non-local patch method [17], and propose an improved color patch similarity measure (ICPSM) to compute the color measure from three aspects: (1) we introduce a patch manner to compute the color similarity and apply a non-linear median filter function to replace the general linear Gaussian function to reduce blurring; (2) we construct a replicated patch that centered at the noisy pixel (i, j) to substitute the normal patch to reject noise more effectively; (3) we calculate the smoothing parameter of the proposed ICPSM adaptively based on the noise degree of the input image to steer smoothing.

Our paper is organized as follows: Sect. 2 introduces the WMF based optical flow method. Section 3 describes our proposed improved color patch similarity measure. We describe the implementation detailed in Sect. 4. Experimental comparisons are shown in Sect. 5. A brief conclusion is given in Sect. 6.

2 Weighted Median Filter Based Optical Flow Method

Median filtering [11,12] the intermediate flow fields during incremental estimation and warping is effective to remove outliers. However, it also over-smoothes corners, edges or thin structures. To prevent this kind of over-smoothing, Sun *et al.* [5] construct a weighted non-local term (a practical WMF). We use the WMF based variational optical flow algorithm [5] as a baseline method, and its objective function is expressed as:

$$\begin{aligned} E(\mathbf{u},\mathbf{v},\widehat{\mathbf{u}},\widehat{\mathbf{v}}) &= \sum_{i,j} \rho_D(\mathrm{I}_2(i+u_{i,j}, j+v_{i,j}) - \mathrm{I}_1(i,j)) + \lambda(\rho_S(\| \nabla \mathbf{u} \|) + \rho_S(\| \nabla \mathbf{v} \|)) \\ &+ \lambda'(\| \mathbf{u} - \widehat{\mathbf{u}} \|^2 + \| \mathbf{v} - \widehat{\mathbf{v}} \|^2) \\ &+ \sum_{i,j} \sum_{(i',j') \in N_{i,j}} w_{i,j,i',j'}(|\widehat{u}_{i,j} - \widehat{u}_{i',j'}| + |\widehat{v}_{i,j} - \widehat{v}_{i',j'}|) \end{aligned} \tag{1}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the horizontal and vertical components of the optical flow field that represents the displacements between the input image pair I_1 and I_2 and, λ and λ' are the weighting parameters controlling the relative importance of each term. $\rho_S(x) = \rho_D(x) = (x^2 + \xi^2)^\alpha$ is the slightly non-convex penalty function, with $\alpha = 0.45, \xi = 0.001$. $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{v}}$ are the auxiliary flow fields of $\mathbf{u}$ and $\mathbf{v}$, and approximate to them. (i', j') is the spatial position of any pixel that belongs to a neighborhood $N_{i,j}$ of pixel $(i,\, j)$.

$w_{i,j,i',j'}$ is the weighting function of the last weighted non-local term, it denotes the similarity between pixel $(i,\, j)$ and its neighborhood pixels. $w_{i,j,i',j'}$ gives high values to pixels belonging to the same surface, while it gives low values to pixels corresponding to corners, edges and thin structures. It is calculated according to spatial distance and color similarity between pixels, and the occlusion state:

$$w_{i,j,i',j'} \propto \exp\{-(Spa(i,j,i',j') + Col(i,j,i',j'))\} \frac{O(i',j')}{O(i,j)} \tag{2}$$

in particular, the spatial distance measure is defined as:

$$Spa(i,j,i',j') = \frac{|i-i'|^2 + |j-j'|^2}{2\sigma_S^2} \tag{3}$$

the color similarity is defined as:

$$Col(i,j,i',j') = \frac{|\mathrm{I}(i,j) - \mathrm{I}(i',j')|^2}{2\sigma_C^2} \tag{4}$$

where $\mathrm{I}(i, j)$ and $\mathrm{I}(i', j')$ are the color vectors in the CIELab space of the central pixel $(i,\ j)$ and its neighborhood pixel (i', j') respectively, $\sigma_S = 7$ and $\sigma_C = 7$.

The occlusion state $O(i, j)$ is computed by considering both the flow divergence and pixel projection difference:

$$O(i, j) = \exp\{-\frac{d^2(i, j)}{2\sigma_d^2} - \frac{e^2(i, j)}{2\sigma_e^2}\} \tag{5}$$

where $\sigma_d = 0.3$ and $\sigma_e = 20$. In particular, $d(i, j)$ is the one-sided flow divergence function, defined as:

$$d(i, j) = \begin{cases} \mathrm{div}(i, j), & \mathrm{div}(i, j) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

in which the flow divergence $\mathrm{div}(i, j)$ is computed as:

$$\mathrm{div}(i, j) = \frac{\partial}{\partial x} u(i, j) + \frac{\partial}{\partial y} v(i, j) \tag{7}$$

where $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ are respectively the horizontal and vertical flow derivatives.

The pixel projection difference $e(i, j)$ is defined as:

$$e(i, j) = \mathrm{I}(i, j) - \mathrm{I}(i + u_{i,j}, j + v_{i,j}) \tag{8}$$

Among the three cues, color similarity plays a most significant role [5]. However, as Rashwan *et al.* [13] stated: the reliance on color similarity of the non-local term causes it is affected by noisy pixels, resulting in inaccurate flow vectors and blurred motion boundaries. In the next section, we will describe a patch based color similarity measure to handle the problem of WMF.

3 Improved Color Similarity Measure

We now explain how to improve the color similarity measure by integrating the non-local patch strategy. In particular, we will first introduce a general patch manner [14] based color similarity measure. Afterwards, we describe an improved replicated patch which has better performance to reject noise. Thirdly, we propose an adaptive scheme to select the smoothing parameter to control denoising.

3.1 Color Patch Similarity Measure (CPSM)

After Buades *et al.* [14] proposed an NLM algorithm, the non-local denoising methods have drawn significant attention. The primary advantage of the NLM denoising method is that they utilize patches instead of pixels to calculate intensity similarity, which makes the NLM methods more robust to pixel-based filters. According to this fact, as shown in Fig. 1, we employ the between-patch manner to replace the between-pixel way to compute the color similarity to reduce the influence of the noisy pixels:

$$PCol(i,j,i',j') = \frac{\| \mathbf{PI}(i,j) - \mathbf{PI}(i',j') \|^2}{2\sigma_{PC}^2} \tag{9}$$

where $\mathbf{PI}(i,j)$ and $\mathbf{PI}(i',j')$ denote the local image patches of size $k \times k$ (we set k = 3 in this paper) centered at pixel $(i,\ j)$ and (i',j') respectively. $\|\mathbf{PI}\|$ is the Euclidean norm of patch $\mathbf{PI}$ as a point in $\mathbf{R}^{k^2}$.

3.2 Improved Color Patch Similarity Measure (ICPSM)

With the above modification, the patch difference is used to substitute the pixel difference. For example, for a noisy pixel (i,j), the modification of $PCol(i,j,i',j')$ is expressed as (a 3 × 3 patch):

$$\frac{\sum_{p\in P, q\in P} \| \mathrm{I}(i+p,j+q) - \mathrm{I}(i'+p,j'+q) \|^2}{2\sigma_{PC}^2} \rightarrow \frac{|\mathrm{I}(i,j) - \mathrm{I}(i',j')|^2}{2\sigma_C^2} \tag{10}$$

where $P = [-1,0,1]$. However, the traditional patch measure stills has some defects, in this paper, we improve the CPSM from following three aspects.

PColWMF. Greenberg and Kogan [19] stated that applying a non-linear median filter function rather than a linear Gaussian function can produce less blurring during denoising and make the filter more robust to noise. Based on the idea of [19], we can improve the CPSM $PCol(i,j,i',j')$ as following:

$$PCol(i,j,i',j') = \mathrm{median}(\mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_5, \ldots, \mathrm{ID}_9) \tag{11}$$

the color difference $\mathrm{ID}_n (n = 1,2,\ldots,9)$ is computed as:

$$\mathrm{ID}_n = \frac{|\mathrm{I}(i+p,j+q) - \mathrm{I}(i'+p,j'+q)|^2}{2\sigma_{PC}^2} \tag{12}$$

where $p \in P, q \in P$. In particular, $\mathrm{ID}_1 = \frac{|\mathrm{I}(i-1,j-1)-\mathrm{I}(i'-1,j'-1)|^2}{2\sigma_{PC}^2}$, ..., $\mathrm{ID}_5 = \frac{|\mathrm{I}(i,j)-\mathrm{I}(i',j')|^2}{2\sigma_{PC}^2}$, ..., $\mathrm{ID}_9 = \frac{|\mathrm{I}(i+1,j+1)-\mathrm{I}(i'+1,j'+1)|^2}{2\sigma_{PC}^2}$.

RPColWMF. The traditional patch based color measure Eq. (9) focuses on comparing the similarity between the patch $\mathbf{PI}(i,j)$ and its neighboring patches $\mathbf{PI}(i',j')$, and the patches that belong to the same surface will be given higher weight during filtering. However, in each $\mathbf{PI}(i,j)$, the central pixel $(i,\ j)$ is the most significant, and its neighborhood pixels are not so important. For example, as shown in Fig. 1(b), if one (or more) neighboring pixel of $(i,\ j)$ in $\mathbf{PI}(i,j)$ is an outlier (like the red point), the correctness of the similarity between $\mathbf{PI}(i,j)$ and $\mathbf{PI}(i',j')$ is badly violated. Colors in natural images are locally consistent, one pixel has a very large chance of being similar to some of its neighbors [22]. That is to say, one non-noise pixel should be similar to some of its neighboring patches.

According to this characteristic, to overcome the above mentioned problem, we replicate the central pixel (i, j) to construct a new patch $\mathbf{RPI}(i, j)$ (see Fig. 1(c)) and compute the color similarity between the replicated patch $\mathbf{RPI}(i, j)$ and its neighboring patches $\mathbf{PI}(i', j')$, which can be expressed as $RPCol(i, j, i', j')$:

$$RPCol(i, j, i', j') = \mathrm{median}(\mathrm{RID}_1, \mathrm{RID}_2, \ldots, \mathrm{RID}_5, \ldots, \mathrm{RID}_9) \tag{13}$$

where $\mathrm{RID}_n = \frac{|\mathrm{I}(i,j)-\mathrm{I}(i'+p,j'+q)|^2}{2\sigma_{PC}^2}$, $(p \in P, q \in P)$.

This improvement ensures that if the color of a pixel (i, j) is approximate to the color of its neighboring patch $\mathbf{PI}(i', j')$, a large value will be assigned to $RPCol(i, j, i', j')$, if not, the value of $RPCol(i, j, i', j')$ is small. Clearly, this strategy is very effective to remove noisy pixels as nearly no noise can have a similar color to an image patch.

Smoothing Parameter Selection. Different from [5] which just set the color smoothing parameter $\sigma_C = 7$ fixed for all sequences, we calculate the σ_{PC} according to:

$$\sigma_{PC} = 9(1 + \log_{10}(\sigma)) \tag{14}$$

where σ is the standard deviation of the noise of the input color image I, and under a constraint $\sigma = max(\sigma, 1/10)$. The noise is computed like this: during the coarse-to-fine optimization framework, we use the Gaussian filter to pre-filter I before downsampling it at each scale, to construct a pyramid of images [1]. Hence, at each pyramid level, we consider noise as the difference of the denoised and the noisy color image I.

This scheme is helpful to modify the denoising performance due to two advantages: (1) in contrast to the fixed manner [5], it adjusts the smoothing parameter σ_{PC} accord with the noise degree of the input image I. Since the noise degree between different images is completely different, a fixed smoothing parameter is not suitable for all kinds of images; (2) comparing to the non-local filters [15,16] which select the smoothing parameter based on $\sigma_{PC} = 10\sigma$, our scheme computes the smoothing parameter more precise. For heavy noisy image the smoothing parameter will be enlarged, while for low noisy image the smoothing parameter will be decreased. This principle satisfies the basic denoising feature. Table 2 demonstrates the effectiveness of this scheme.

Due to the ICPSM, our modified weighted non-local term — we refer to it as PatchWMF, is much more robust to the noisy pixels. The boundary blurring is reduced and the accuracy of the estimated flow field is modified.

4 Implementation

We follow the optimization framework of [5] to compute the flow field, more importantly, some useful practices are used for further modification.

Edge-Preserving Smoothness. To preserve edges, we redefine the smoothness term as [9]:

$$E(\mathbf{u}, \mathbf{v}) = \sum_{i,j} \omega(i, j)(\rho_{\mathrm{S}}(\| \nabla \mathbf{u} \|) + \rho_{\mathrm{S}}(\| \nabla \mathbf{v} \|)) \tag{15}$$

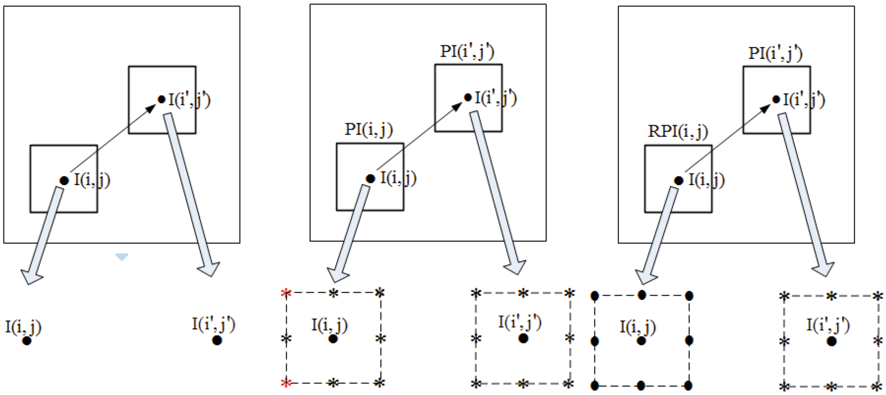

Fig. 1. Three color weight measures. **From Left to Right**: (a) Pixel measure. (b) Patch measure. (c) Replicated patch measure (Color figure online).

the edge-preserving term $\omega(i,j)$ is a structure adaptive map which contains motion discontinuity [9,10]:

$$\omega(i,j) = \exp(-\|\nabla \mathrm{I}_1\|^k) \tag{16}$$

where we set $k = 0.8$.

Large Displacements Handling. The traditional coarse-to-fine framework is good at estimating large displacements of relatively large objects, however, it performs poorly on fine scale structures with motions larger than their size. That because fine scale structures may disappear in coarse scales, leading to no valid matching information in the coarse level can be propagated to finer scale, hence, these structures cannot be well recovered. To handle large displacements while preserve motion details, similar as [10], we first use approximate nearest neighbor fields (NNF) to compute an initial dense correspondence field. Furthermore, we employ the SIFT feature detection and selection method of [9] to obtain some reliable sparse matches. Then, we incorporate the dense NNF and the sparse SIFT matches through the Quadratic Pseudo-Boolean Optimization (QPBO) fusion method [20] to get an improved NNF. After that, during the coarse-to-fine optimization, before the first warping step on each pyramid level, we fuse the improved NNF and the coarser lever computed continuous flow field ($\mathbf{u}$, $\mathbf{v}$) as flow initialization.

Occlusion Detection and Post-processing. Occlusion detection is a notoriously difficult problem, since displacement vectors for occluded pixels generally cannot be determined due to the lack of correspondences. Current optical flow models are not yet powerful enough to handle this problem, thus it is beneficial to tackle occlusions in the post-processing. We employ the mapping uniqueness criterion of [21] to detect occlusions, and the occlusion state is expressed as:

$$Occ(i,j) = \min(\frac{\max(N(i+u_{i,j}, j+v_{i,j})-1, 0)}{2}, 1) \tag{17}$$

where $N(i+u_{i,j}, j+v_{i,j})$ denotes the number of pixels in the reference image I_1 that corresponds to a pixel located at $(i+u_{i,j}, j+v_{i,j})$ in the target image I_2. We regard the pixel as occluded pixel when $Occ(i,j) \geq 0.5$. To remove these artifacts while preserving the object boundaries, we apply the joint bilateral filter to fill the detect occluded pixels.

For numerical calculation, we adopt the 3-stage Graduated Non-Convexity (GNC) scheme and perform 5 warping steps on each pyramid level at GNC stage 2. At other GNC stages, due to large motions require more warping iterations [5], we perform 10 warping steps if the sequence is dominated by large displacements, if not we perform 5 warping steps. Other implementation methods and optimization steps are same as [5]. All experiments are performed on a Laptop with an Intel Core i5-2410M 2.30 GHz processor and 4 GB memory. Regarding the running time, in our current CPU implementation, the whole program takes 520 s to compute a high quality flow field for an image pair with resolution 640480 in, for instance, the Urban sequence.

5 Experiments

In this section, we test our method, referred to as PatchWMF-OF, on three public challenging optical flow benchmarks the Middlebury benchmark [3], the MIT benchmark [24], and the MPI Sintel benchmark [23]. Both quantitative comparison in terms of two standard error measures the average angular error (AAE) and the average endpoint error (EPE), and visual analysis compares with other related techniques are performed.

5.1 Evaluation of PatchWMF Technique

We evaluate the proposed PatchWMF technique by testing whether the three presented schemes that aim to improve the color similarity measure are effective, and by quantitatively comparing it with the baseline methods – WMF [5]. In particular, the PatchWMF technique is a combination of the RPColWMF approach and the smoothing parameter selection (Eq. (14)) strategy. Table 1 shows the AAE and EPE results on 8 synthetic sequences from the Middlebury training set. The error statistics display that the three schemes are useful, leading to the PatchWMF outperforms the WMF.

5.2 Evaluation of Noisy Pixels Handling

To further evaluate the noisy pixels handling ability of our PatchWMF technique, we synthesize the 8 training sequences from the Middlebury benchmark by adding Gaussian noise with variance $\sigma_n = [10, 20, 30]$ respectively. From Table 2, we can see that the proposed PatchWMF performs much better than the WMF. Comparing Table 2 with Table 1, it is clear that for a same sequence but with different noise level, the motion estimation accuracy improvement (i.e. AAE and EPE modification) obtained from our PatchWMF on the noisy

Table 1. AAE/EPE of the eight training sequences from the Middlebury benchmark with different color measures for the WMF

Method	WMF($\sigma_C = 7$)	PColWMF($\sigma_{PC} = 7$)	RPColWMF($\sigma_{PC} = 7$)	PatchWMF
RubW.	2.351/0.073	**2.253**/**0.071**	2.429/0.074	2.400/0.072
Venus	3.327/0.237	3.421/0.243	3.363/0.238	**3.227**/**0.232**
Dime.	2.570/0.131	2.595/0.132	2.586/0.132	**2.380**/**0.121**
Hydra.	1.829/0.151	**1.785**/**0.145**	1.866/0.155	1.837/0.153
Urban2	2.082/0.221	2.095/0.227	2.073/**0.218**	**2.054**/0.221
Urban3	2.600/0.394	2.754/0.391	**2.583**/0.388	2.660/**0.387**
Grove2	1.498/0.104	1.601/0.112	1.509/0.104	**1.433**/**0.099**
Grove3	4.955/0.463	5.079/0.478	4.957/0.462	**4.829**/**0.460**
Avg.	2.652/0.222	2.700/0.225	2.670/0.221	**2.600**/**0.218**

Table 2. Comparison of the AAE and EPE of the Middlebury training sequences with different added Gaussian noise level for the WMF/PatchWMF techniques

Method	$\sigma_n = 10$		$\sigma_n = 20$		$\sigma_n = 30$	
	AAE	EPE	AAE	EPE	AAE	EPE
RubW.	5.122/**4.975**	0.154/**0.148**	8.703/**7.899**	0.262/**0.237**	11.97/**10.89**	0.351/**0.323**
Venus	4.245/**3.953**	0.302/**0.282**	5.423/**5.238**	0.421/**0.406**	**7.197**/7.294	0.546/**0.535**
Dime.	2.819/**2.615**	0.140/**0.131**	4.574/**4.251**	0.229/**0.214**	7.183/**7.122**	0.347/**0.340**
Hydra.	2.379/**2.299**	0.214/**0.208**	3.482/**3.202**	0.328/**0.303**	4.299/**3.984**	0.423/**0.390**
Urban2	2.945/**2.845**	**0.276**/0.279	**4.387**/4.447	**0.386**/0.393	6.286/**6.142**	0.514/**0.508**
Urban3	**3.778**/3.817	**0.475**/0.480	**5.325**/5.387	0.637/**0.630**	7.971/**7.965**	0.872/**0.870**
Grove2	1.751/**1.697**	0.126/**0.120**	2.275/**2.185**	0.163/**0.155**	3.211/**3.101**	0.231/**0.223**
Grove3	5.521/**5.520**	0.525/**0.521**	**6.243**/6.249	**0.608**/0.610	7.333/**7.295**	**0.688**/0.698
Overall improve	2.95 %	1.95 %	3.85 %	2.83 %	3.00 %	2.25 %

Table 3. AAE/EPE of two noisy training sequences from the MIT benchmark [24]

Method	Cameramotion		Fish	
	AAE	EPE	AAE	EPE
WMF	6.408	0.566	26.109	0.731
PatchWMF	6.285	0.550	25.064	0.692
Improve	2.0 %	2.8 %	4.0 %	5.4 %

part is much higher than on the clean part. For example, for the original RubberWhale sequence (without adding noise), the AAE/EPE of the PatchWMF is approximate to the AAE/EPE of the WMF; in contrast, for the noise added RubberWhale sequences, the AAE/EPE of the PatchWMF is significantly decreased compare to the corresponding AAE/EPE of the WMF. The AAE improvement of $\sigma_n = 20$ and $\sigma_n = 30$ is about 3 %. The results well demonstrate the effectiveness of our ICPSM, and making our PatchWMF is much more robust to against noise than the WMF. Additionally, Figs. 2 and 3 show two visual comparison.

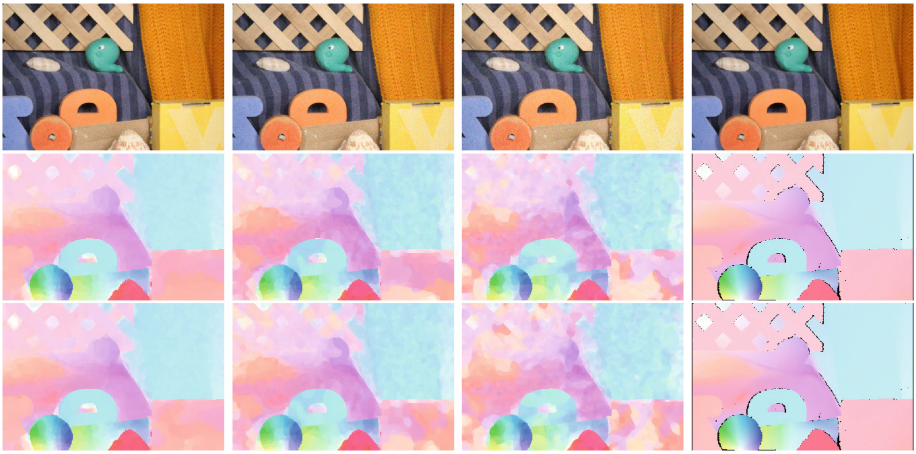

Fig. 2. Visual comparison on the Gaussian noise added RubberWhale sequence [5]. **From Top to Bottom:** RubberWhale sequences, flow results of WMF, flow results of PatchWMF. **From Left to Right:** flow with added Gaussian noise $\sigma_n = 10$, $\sigma_n = 20$, $\sigma_n = 30$, and the ground truth flows.

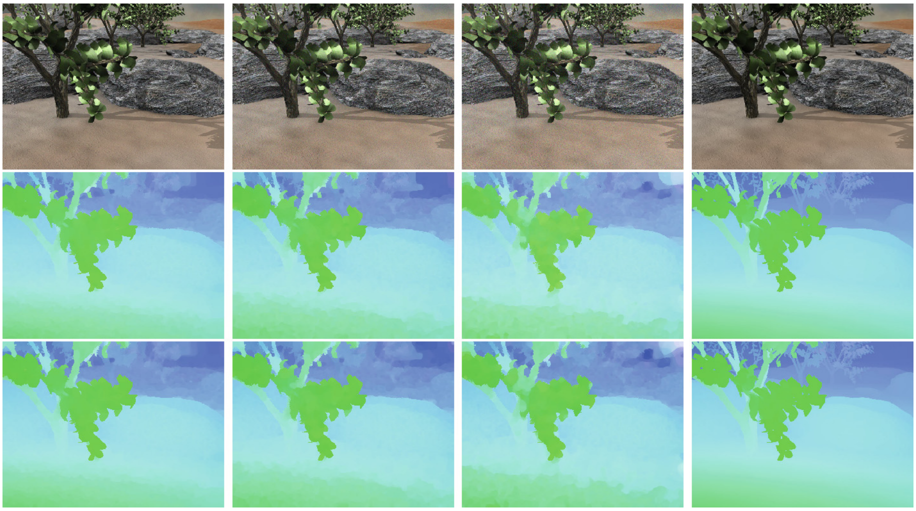

Fig. 3. Visual comparison on the Gaussian noise added Grove2 sequence [5]. **From Top to Bottom:** Grove2 sequences, flow results of WMF, flow results of PatchWMF. **From Left to Right:** flow with added Gaussian noise $\sigma_n = 10$, $\sigma_n = 20$, $\sigma_n = 30$, and the ground truth flows.

The motion boundaries of our flow fields are more accurately preserved. In contrast, due to the disturbance of the noisy pixels, the WMF fails to recover edges, resulting in a lot of errors that are produced by motion blurring distribute at edge regions.

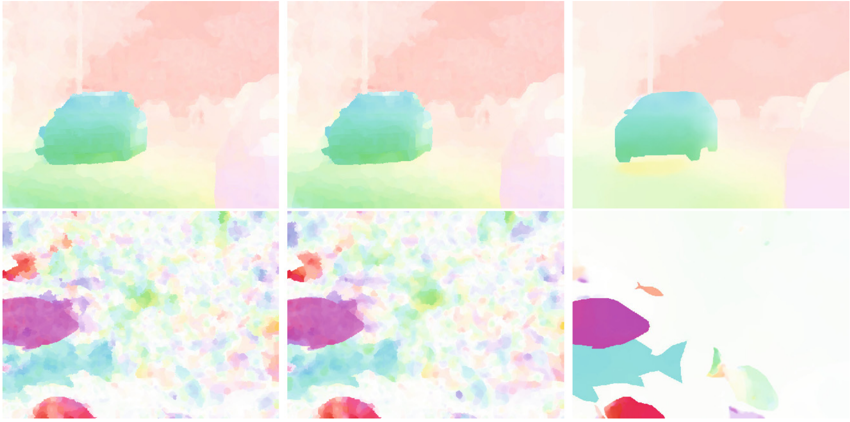

Fig. 4. Visual comparison on two heavy noisy real-world sequences from the MIT benchmark [24]. **From Top to Bottom:** estimated flow fields of cameramotion, estimated flow fields of fish. **From Left to Right:** estimated flow fields with WMF, estimated flow fields with PatchWMF, and the ground truth flows.

Quantitative comparison (Table 3) and visual comparison (Fig. 4) of the denoising results of the WMF and our PatchWMF according to two heavy noisy sequences – cameramotion and fish are conducted. Table 3 reveals that our PatchWMF is superior to WMF to obtain more accurate motion field. In Fig. 4, since the two sequences are fully filled with noisy pixels, the WMF is badly violated and it performs poor on object boundaries. Contrastively, the presented ICPSM helps the WMF robustly reject outliers, thus the motion blurring is reduced in our estimated flow fields.

5.3 Results on MPI Sintel Benchmark

To evaluate the overall performance of our PatchWMF-OF method, we test it on the challenging MPI Sintel benchmark [23]. This benchmark contains long photo-realistic video sequences with extremely difficult cases, e.g., large motions, specular reflections, motion blur, defocus blur, and atmospheric effects. The evaluation is conducted on two kinds of frames, namely clean pass and final pass. EPE all measures the EPE over all pixels, and s0–10 measures pixels with a speed between 10 and 40 pixels (similarly for s10–40 and s40+). Tables 4 and 5 compare our method to state-of-the-art algorithms on the test set of the MPI Sintel benchmark. At the time of submission, for EPE all, it is ranked 7th on the clean pass and 11th on final pass; while for s0–10, it is ranked 1th on the clean pass and 7th on final pass. More importantly, it outperforms current published methods (the huge memory consumption of the DeepFlow [25] prevents itself from practical applications, thus it should be rule out). The results illustrate that our method performs topmost for both large and small displacement optical flow estimation in a unified framework, especially for small motion (i.e. s0–10).

Table 4. Clean pass results on the MPI Sintel test set

Method	EPE all	EPE s0–10
DeepFlow [25]	**5.377**	0.960
PatchWMF-OF	5.550	**0.581**
MDP-Flow2 [9]	5.837	0.640
EPPM	6.494	1.402
S2D-Matching	6.510	0.622
Classic+NLP [5]	6.731	0.638
MLDP-OF	7.297	0.600
LDOF	7.563	0.936

Table 5. Final pass results on the MPI Sintel test set

Method	EPE all	EPE s0–10
DeepFlow [25]	**7.212**	1.284
S2D-Matching	7.872	**1.172**
PatchWMF-OF	7.971	1.279
MLDP-OF	8.287	1.312
Classic+NLP [5]	8.291	1.208
EPPM	8.377	1.834
MDP-Flow2 [9]	8.445	1.420
LDOF	9.116	1.485

In contrast to the baseline method [5], our algorithm performs much better on estimating large motions, which demonstrates our large displacements handling scheme is effective and necessary. On the other side, our method also performs better on small motion estimation. For the clean pass, the EPE of s0–10 is reduced from 0.638 to 0.581, the improvement is about 10 %; for the final pass, the EPE of s0–10 is changed from 1.208 to 1.279, nearly the same. Why our method performs worse on the final pass than on the clean pass? Since the final pass is rendered with motion blur, defocus blur and atmospheric effects while the clean pass are not, and the motion blur and defocus blur do not affect the accuracy of the results too much [26], thus the reason for degradation is due to the synthetic atmospheric effects, not because of noisy pixels. This fact indicates that our ICPSM based WMF is superior to the WMF [5] for rejecting noisy pixels, and outliers can be removed more accurately. Figure 5 shows representative results of the final pass sequences. It is easy to find that our method preserves edges and motion boundaries better than other two related algorithms [5,9], and it also well captures both large and small motions. In particular, comparing to [5], boundary blurring due to noisy pixel perturbation is reduced.

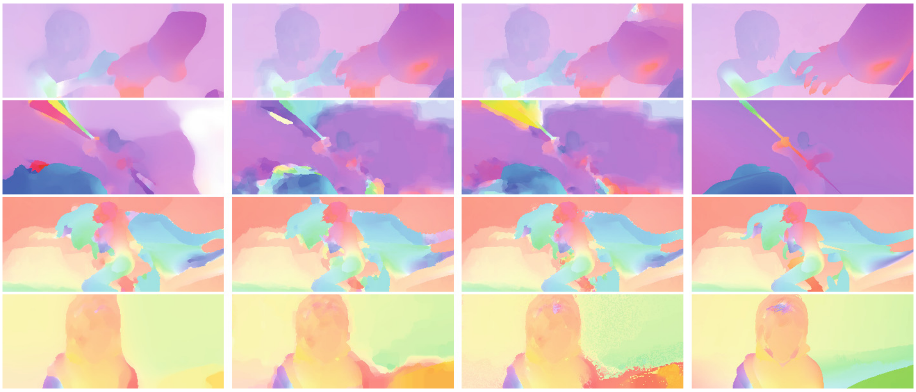

Fig. 5. Visual comparison on the final pass of the MPI Sintel test set. **From Top to Bottom:** shaman1, ambush3, cave3, wall. **From Left to Right:** MDP-Flow2 [9], Classic+NLP [5], PatchWMF-OF(Ours), ground truth flows.

6 Conclusion

This paper addresses the problem of flow field outliers removing depends on the median filter. We present an improved color patch similarity measure to modify the robustness of the WMF to noise. By using the patch difference to replace the pixel difference, the violation due to the noisy pixels is reduced. Additionally, to further improve the noise rejection performance of the patch scheme, a replicated patch method is proposed. Moreover, we introduce an adaptive smoothing parameter selection method to calculate the appropriate smoothing parameter according to the noise degree of the input image. Experiments in this paper have demonstrated that the improved color patch similarity measure is effective to reduce the noise affection – the color difference between a noisy pixel and its neighboring pixel may similar, but the color difference between a replicated patch of a noisy pixel and a patch of its neighboring pixel cannot be similar, leading to the improved WMF denoises intermediate flow fields more accurately.

Acknowledgements. This publication was supported by the Dutch national program COMMIT.

References

1. Horn, B., Schunck, B.: Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
2. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
3. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. Int. J. Comput. Vis. **92**, 1–31 (2011)

4. Black, M.J., Anandan, P.: The robust estimation of multiple motions: parametric and piecewise smooth flow elds. Comput. Vis. Image Underst. **63**, 75–104 (1996)
5. Sun, D., Roth, S., Black, M.J.: A quantitative analysis of current practices in optical flow estimation and the principles behind them. Int. J. Comput. Vis. **106**, 115–137 (2014)
6. Xiao, J., Cheng, H., Sawhney, H.S., Rao, C., Isnardi, M.: Bilateral filtering-based optical flow estimation with occlusion detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 211–224. Springer, Heidelberg (2006)
7. Tu, Z., Aa, N., Gemeren, C.V., Veltkamp, R.C.: A combined post-filtering method to improve accuracy of variational optical flow estimation. Pattern Recogn. **47**, 1926–1940 (2014)
8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
9. Xu, L., Jia, J., Matsushita, Y.: Motion detail preserving optical flow estimation. IEEE Trans. Pattern Anal. Mach. Intel. **16**, 1744–1757 (2012)
10. Chen, Z., Jin, H., Lin, Z., Cohen, S., Wu, Y.: Large displacement optical flow from nearest neighbor fields. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2443–2450 (2013)
11. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for TV-L^1 optical flow. In: Cremers, D., Rosenhahn, B., Yuille, A.L., Schmidt, F.R. (eds.) Statistical and Geometrical Approaches to Visual Motion Analysis. LNCS, vol. 5604, pp. 23–45. Springer, Heidelberg (2009)
12. Wedel, A., Pock, T., Cremers, D.: Structure- and motion adaptive regularization for high accuracy optic flow. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1663–1668 (2009)
13. Rashwan, H.A., Garca, M.A., Puig, D.: Variational optical flow estimation based on stick tensor voting. IEEE Trans. Image Process. **22**, 2589–2599 (2013)
14. Buades, A., Coll, B., Morel, J.: A review of image denoising algorithms, with a new one. SIAM Multiscale Model. Simul. **4**, 490–530 (2005)
15. Chaudhury, K.N., Singer, A.: Non-local Euclidean medians. IEEE Sig. Process. Lett. **19**, 745–748 (2012)
16. Sun, Z., Chen, S.: Analysis of non-local Euclidean medians and its improvement. IEEE Sig. Process. Lett. **20**, 303–306 (2013)
17. Aharon, M., Elad, M., Bruckstein, A.M.: The K-SVD: an algorithm for designing of over complete dictionaries for sparse representation. IEEE Sig. Process. Lett. **54**, 4311–4322 (2006)
18. Chatterjee, P., Milanfar, P.: Patch-based near-optimal image denoising. IEEE Trans. Image Process. **21**, 1635–1649 (2012)
19. Greenberg, S., Kogan, D.: Improved structure-adaptive anisotropic filter. Pattern Recognit. Lett. **27**, 59–65 (2006)
20. Rother, C., Kolmogorov, V., Lempitsky, V.S., Szummer, M.: Optimizing binary MRFS via extended roof duality. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
21. Kim, T.H., Lee, H.S., Lee, K.M.: Optical flow via locally adaptive fusion of complementary data costs. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3344–3351 (2013)
22. Zhang, Q., Xu, L., Jia, J.: 100+ times faster weighted median filter (WMF). In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2014)

23. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 611–625. Springer, Heidelberg (2012)
24. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
25. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: DeepFlow: large displacement optical flow with deep matching. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1385–1392 (2013)
26. Bao, L., Yang, Q., Jin, H.: Fast edge-preserving patchmatch for large displacement optical flow. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2014)