Pattern Recognition 47 (2014) 1926–1940

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A combined post-filtering method to improve accuracy of variational optical flow estimation

Zhigang Tu, Nico van der Aa, Coert Van Gemeren, Remco C. Veltkamp*

Department of Information and Computing Sciences, Utrecht University, Utrecht, Netherlands

ARTICLE INFO

Article history:
Received 11 July 2013
Received in revised form
17 October 2013
Accepted 25 November 2013
Available online 11 December 2013

Keywords:
Optical flow
Combined post-filtering (CPF)
Multi-scale nonlinear 3D structure tensor
Hybrid GBF and Gaussian Filter smoothing (HGBGF)
Spatial-scale gradient signal-to-noise ratio (SNR)

ABSTRACT

We present a novel combined post-filtering (CPF) method to improve the accuracy of optical flow estimation. Its attractive advantages are that outliers reduction is attained while discontinuities are well preserved, and occlusions are partially handled. Major contributions are the following: First, the structure tensor (ST) based edge detection is introduced to extract flow edges. Moreover, we improve the detection performance by extending the traditional 2D spatial edge detector into spatial-scale 3D space, and also using a gradient bilateral filter (GBF) to replace the linear Gaussian filter to construct a multi-scale nonlinear ST. GBF is useful to preserve discontinuity but it is computationally expensive. A hybrid GBF and Gaussian filter (HGBGF) approach is proposed by means of a spatial-scale gradient signal-to-noise ratio (SNR) measure to solve the low efficiency issue. Additionally, a piecewise occlusion detection method is used to extract occlusions. Second, we apply a CPF method, which uses a weighted median filter (WMF), a bilateral filter (BF) and a fast median filter (MF), to post-smooth the detected edges and occlusions, and the other flat regions of the flow field, respectively. Benchmark tests on both synthetic and real sequences demonstrate the effectiveness of our method.

1. Introduction

Motion is an intrinsic characteristic of the world [1], providing essential information that can be used in a wide variety of image processing and visual tasks, such as 3D-reconstruction, segmentation, tracking and video compression. One of the most successful motion estimation approaches is the variational optical flow method [2,3], due to two inherent advantages, i.e. its comprehensive functional form and an efficient numerical optimization. The variational optical flow method was introduced by Horn and Schnuck (HS) [4]. It combines a local, gradient-based data matching term with a global smoothness term. The data term assumes that each pixel's brightness remains invariant during a short time. The smoothness term regularizes each pixel's flow by its neighbors' flow. It assumes that the flow vector varies smoothly almost everywhere over the flow field. In practice, however, these two basic constraints are seriously violated. Various extensions and improvements have been proposed during the past 30 years in order to overcome the drawbacks of the original HS model.

These variations can generally be classified according to the following three aspects: (1) Modification of the variational formulation, such as improvements of the data term to make the algorithm more robust under illumination changes [5], invariant under different types of motion [6], and more resistant to noise [7] and outliers [6,7], and large displacements[8]. Modifications of the regularizer's capability to handle motion discontinuity [9,10]. Selection of the optimal weighting parameter λ to obtain a better balance between the data term and the regularization term [10,11]. (2) Pre-processing of the input frames to reduce violations, such as noise suppression of the frames to remove, for example, high frequencies that might have a negative influence on the result. Commonly used filtering methods include Gaussian filter [4], PDE filter [12] and non-local filter [13]. Most of these methods do not only reduce noise, but also enhance important structures of the frames [14]. (3) Post-processing of the flow field to improve the accuracy, e.g. usage of available filters for smoothing, such as, Kalman filter [15], median filter (MF) [14,16], and bilateral filter (BF) [17,18].

Wedel et al. [16] successfully introduced a MF to remove the flow noise. However, his MF approach over-smoothes the edges. Sun et al. [14] proposed a modified WMF method to prevent this kind of over-smoothing, and saved the computational time by solely smoothing the detected motion boundaries with the Sobel edge detector. However, this method still has some drawbacks. First, the Sobel detector often performs poorly in extracting flow boundaries. Second, wrong flow components in the MF [19] window can cause serious errors. Surprisingly, although smoothing the flow field

* Corresponding author. Tel.: +31 30 253 4091.
E-mail addresses: Z.Tu@uu.nl (Z. Tu), N.vanderAa@noldus.nl (N. van der Aa), C.J.Vangemeren@uu.nl (C. Van Gemeren), R.C.Veltkamp@uu.nl (R.C. Veltkamp).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.11.026

boundaries is a reasonable way to improve accuracy and efficiency, few efforts have been devoted so far to analysis of the connection between the smoothing performance and the extraction of flow features (e.g. edges and occlusions). To the best of our knowledge, this work is the first systematical analysis of the importance of the above mentioned connection.

We present a novel 3D nonlinear ST based Harris edge detector to identify flow edges, and apply a piecewise occlusion detection approach to detect flow occlusions. The ST has first been proposed by Förstner and Gülch [20]. Since it represents the first order derivative information of an image, it can be used as a local geometry indicator to analyze the geometric structure of a scalar-valued data set (e.g. an image) or a vector-valued data set (e.g. the flow field). Compared to traditional derivative-based methods, the ST has two outstanding characteristics due to two Gaussian smoothing operations: (1) smoothing the data set yields robustness under noise by introducing an integration scale and (2) integrating local structure information (e.g. orientation) from a neighborhood makes ST able to distinguish features [21].

Since Gaussian smoothing is isotropic, it has some disadvantages: (1) detailed and weak features, such as some textures are smoothed out, (2) distinctive discontinuities such as edges are blurred and, and (3) points belonging to different regions, such as occluded points and non-occluded points, would be roughly composed. These disadvantages are caused by the fact that the Gaussian filter is fixed in both size and shape, and it cannot adapt to the local structures. Therefore, the Gaussian filter based linear ST cannot detect edges accurately. For instance, the identified edges are often wider than the real edges or discontinuous.

Different anisotropic filtering methods have been proposed to replace the linear Gaussian filter to construct a nonlinear ST, like anisotropic diffusion [21], BF [17,18,22] and mean shift filtering [23]. They can adapt to local structures, avoiding smoothness across discontinuities and preserving useful information. The BF, which extends the concept of Gaussian filter by adding a Gaussian weighting function that depends on the difference between pixel intensities, is most attractive [24] due to its inherent advantages: (1) it is non-iterative, which makes it overcomes the instability of the iterative method – since small errors in derivatives will be magnified after each iteration, (2) only two parameters are needed and these parameters have explicit geometrical and graphical meaning, therefore, they are easy to be constructed and implemented, and (3) as illustrated in [17], the BF can handle occlusion. In this work, the BF is used to replace the Gaussian filter to construct a nonlinear ST, and also it is used to replace the MF to smooth the occlusions of the flow field.

The BF assigns higher weights to pixels with smaller spatial and/or color distances computed with respect to the central pixel. In this way, smoothing is implemented adapt to local structures. To distinguish trivial structures from true corners, Zhang et al. [22] introduced a GBF which uses both spatial and gradient distances to smooth the 2D spatial ST. In this work, we introduce the GBF to construct a nonlinear 3D ST to detect edges.

A direct implementation of the BF is computationally expensive. It requires $O(\sigma_s^2)$ operations per pixel. Especially when the data set is large, it is too slow to be executed in real-time. Porikli [25] proposed a fast O(1) BF using Taylor polynomials to approximate the standard Gaussian BF. However, the Taylor polynomials provide only good approximations of the range Gaussian function just locally around the origin. Different from the O(1) BF method, in this work, we apply a fast spatial-scale gradient based SNR segmentation and a hybrid smoothing approach – HGBGF to treat the low efficiency of the BF.

For a vector-valued data set, the primary derivative errors are concentrated at discontinuities [26]. Because the Gaussian filter is good enough to smooth flat regions and the BF is better at smoothing discontinuities, combing the advantages of the two filters can preserve edges, tackle occlusions and also reduce time consumption. Therefore, we present a novel spatial-scale gradient SNR measure to extract discontinuities. Then, we apply the BF and the Gaussian filter to smooth the ST elements in separate regions, respectively.

Multi-scale is an intrinsic property of the signal structure in nature [27]. Liu et al. [28] gave a definition of edge scale and pointed out that there exists an optimal scale of the edge – the optimal scale is a parameter to indicate at which resolution(s) an edge is most salient for a human. We extend the traditional spatial ST based edge detector into spatial-scale space by adding scale information. Integrating the HGBGF into our spatial-scale 3D ST, a local pattern adaptive framework is constructed, resulting in better detection of flow field edges.

Using a suitable filter, such as the MF, to post-filter the intermediate flow field during incremental estimation and warping is an effective way to remove outliers and a key technique to recent performance gains [14,16]. However, the MF is not good at handling occlusions. In contrast, the BF has been successfully applied to treat occlusion [17,18]. In this paper, we combine the advantages of the WMF [14] and the BF[16], and propose a Combined Post-Filtering (CPF) method to smooth the classified flow field regions.

The paper is organized as follows: Section 2 describes the proposed "Classic+CPF" optical flow algorithm. In Section 3, a nonlinear 3D spatial-scale Harris edge detector to detect flow edges, and a piecewise occlusion detection approach to extract occlusions are introduced. A CPF method for post-filtering different regions of the flow field with different filters is proposed in Section 4. In Section 5, experiments and evaluations are conducted to the proposed algorithm. The paper is concluded in Section 6, which includes possibilities for future development.

2. "Classic+CPF" optical flow algorithm

Based on the brightness constancy assumption (data term), and combined with a global smoothness constraint (smoothness term), Horn and Schunck [4] proposed the variational optical flow method for motion estimation between two successive frames I_1, I_2:

$$E(u,v)==\int_{\Omega}\underbrace{(I_2(x+u,y+v,t+dt)-I_1(x,y,t))^2}_{\text{data term}}\,d\Omega+\lambda\int_{\Omega}\underbrace{(|\nabla u|^2+|\nabla v|^2)}_{\text{smoothness term}}\,d\Omega \tag{1}$$

where $(u,v)=(dx/dt,dy/dt)$ is the displacement vector field. It is a 2D projection of the real 3D motion in the world.

One state-of-the-art variational method is the TV-L1 non-local algorithm – "Classic+NL" [14], which incorporates the WMF during optimization to smooth the flow field. Due to the WMF, the accuracy is significantly improved. However, the WMF is poor to handle occlusions (see Section 4.1). To overcome this problem, we classify the optical flow field into three parts – edge regions, occlusions and flat regions. As illustrated in Section 1, we combine the advantages of WMF and BF, and use a CPF method to smooth flow edges, occlusions as well as flat regions with three different filters. Based on the baseline algorithm of "Classic+NL" [14], a "Classic+CPF" algorithm is proposed:

$$\begin{aligned}E(u,v,\overline{u},\overline{v})=&\sum_{i,j}\{\rho_D(I_1(i,j)-I_2(i+u,j+v))+\lambda_1(\rho_S(|u_x|)+\rho_S(|u_y|)\\&+\rho_S(|v_x|)+\rho_S(|v_y|))\}+\lambda_2(||u-\overline{u}||^2+||v-\overline{v}||^2)\\&+\underbrace{\sum_{i_E,j_E i'_E,j'_E}\ \sum_{\in N_{i_E,j_E}} w_{i_E,j_E,i'_E,j'_{EE}}(|\overline{u}_{i,j}-\overline{u}_{i',j'}|+|\overline{v}_{i,j}-\overline{v}_{i',j'}|)_{|\,i_E,j_E,i'_E,j'_E}}_{\text{edge regions}\rightarrow\text{weighted median Filter}}\end{aligned}$$

$$+\underbrace{\sum_{i_{Occ}j_{Occ}i'_{Occ}j'_{Occ}}\ \sum_{\in N_{i_{Occ}j_{Occ}}} w_{i_{Occ}j_{Occ},ii'_{Occ}j'_{Occ}}(|\overline{u}_{i,j}-\overline{u}_{i',j'}|+|\overline{v}_{i,j}-\overline{v}_{i',j'}|)_{|\,i_{Occ}j_{Occ},i'_{Occ}j'_{Occ}}}_{\text{occlusons}\rightarrow\text{bilateral filter}}$$

$$+\underbrace{\sum_{i_{Flat}j_{Flat}i'_{Flat}j'_{Flat}}\ \sum_{\in N_{i_{Flat}j_{Falt}}} \text{median}(|\overline{u}_{i,j}-\overline{u}_{i',j'}|+|\overline{v}_{i,j}-\overline{v}_{i',j'}|)_{|\,i_{Flat}j_{Flat},i'_{Flat}j'_{Flat}}}_{\text{flat regions}\rightarrow\text{median filter}} \qquad (2)$$

Where $\rho_S(x)=\rho_D(x)=(x^2+\varepsilon^2)^\alpha$, $\alpha=0.45$, and $\varepsilon=0.001$. λ_1 and λ_2 are the weighting parameters, controlling the relative importance of each term. $\overline{u}$ and $\overline{v}$ are the auxiliary flow fields of u and v, and approximations of u and v. $N_{i,j}$ represents the neighborhood of pixel (i, j). The third and the fourth term are weighted non-local terms which integrate the information of the image and the flow field. They impose a particular smoothness assumption within a specified region of the auxiliary flow field ($\overline{u}$, $\overline{v}$) and prevents over-smoothing across discontinuities. $w_{i_E j_E, i'_E j'_E}$ is a WMF parameter of the third non-local term, which is calculated using Eq. (9) in work [14]. $w_{i_{Occ}j_{Occ},i'_{Occ}j'_{Occ}}$ is a BF parameter of the fourth non-local term, which is computed according to (using u for example):

$$w(i,j,i',j')_{|\,u}=\exp\left(-\frac{|i-i'|^2+|j-j'|^2}{2\sigma_1^2}\right)\times\exp\left(-\frac{|I(i,j)-I(i',j')|^2}{2\sigma_2^2}\right)$$
$$\times\exp\left(-\frac{(u_{i,j}-u_{i',j'})^2}{2\sigma_3^2}\right)\times Occ(i,j) \qquad (3)$$

$Occ(i,j)$ is an occlusion function and is computed with Eq. (20). The BF weight of v is calculated in the same way as u. We set $\sigma_1=3.0$, $\sigma_2=7.0$ and $\sigma_3=1.0$.

The optical flow field (u, v) is estimated by separating Eq. (2) into two parts. First, the variational optical flow part, which is used for calculating the flow field (u, v). Second, the post-filtering part, which is used to smooth outliers and obtain an outliers-removed auxiliary flow field ($\overline{u}$, $\overline{v}$). The first part can be solved by the traditional numerical optimization algorithms (e.g. Gauss-Seidel, SOR, and Conjugate Gradient). For the second part, a WMF [14], a BF (Eq. (3)) and a MF [19] are implemented for post-filtering (see Section 4).

3. Flow edges and occlusions detection

Post-filtering the flow field is an effective way to remove outliers [14,16] and improve the accuracy. The flow field is a representation of the apparent motion of each pixel of the input images. Therefore, sharp variations of the flow field (e.g. edges and occlusions) reflect salient changes of the image. Variational optical flow algorithms mostly capture the first-order motion, while they easily fail when sudden motion changes occur. Hence, inaccurate displacement vectors are concentrated at edges – intensities sharply change, and occlusions – natural information disappears. Conversely, much fewer flow errors are distributed at flat regions. Smoothing flat regions only gives us a marginal benefit, while some detailed structure information is smoothed out. Furthermore, the computational time will sharply rise as a result of the increased number of filtering points. We select the Middlebury sequences [2] for testing (see Fig. 1), and compare two groups of results which are derived from the classical "Classic+NL" algorithm [14]: one is obtained by MF with just the Sobel detected edges, another is obtained by MF with the whole flow field. Fig. 1 (a) shows that a full MF has no advantage. For most of the sequences (e.g. Dimetrodon, Hydrangea, Venus, Urban2), the accuracy of MF the full flow field is worse than MF the flow edges. On the other side, from Fig. 1(b), we see that the computational time of full MF is much higher (more than 2 times higher) than MF only the detected flow edges. Consequently, extracting flow edges and occlusions, and handling them properly is an effective way to remedy the variational optical flow algorithms as well as save time consumption.

How to effectively identify edges and occlusions of the flow field is the crucial step for post-filtering. However, it is difficult to design feature detectors which can identify edges or occlusions accurately while not responding to other features. In this section, we will describe an accurate nonlinear 3D edge detector to extract flow edges. In additional, the combined flow divergence and pixel projection difference method [18] is used to detect flow occlusions. More importantly, we improve its performance with a piecewise setting.

3.1. Linear 3D spatial-scale Harris edge detection

In the image domain, edge detectors are concerned with the localization of sharp changes of image intensity and the identification of the physical phenomena, which originate from them. Different from detecting image edges, which can use color, texture or other features, the flow field has few cues, the intensity (each flow vector of u and v can be treated as intensity) related approach is an appropriate choice. Brox et al. [21] illustrated that ST is a well-established tool to analyze structure characteristics of a vector-valued data set. In the following, we will describe a multi-scale nonlinear ST, and derive a 3D spatial-scale Harris edge detector to extract flow edges.

3.1.1. 2D spatial ST

For a matrix-valued data set $M(x,\ y)$, the ST of $M(x,\ y)$, S is defined as the outer product of gradient vector $\nabla M\cdot\nabla M^T$ with a

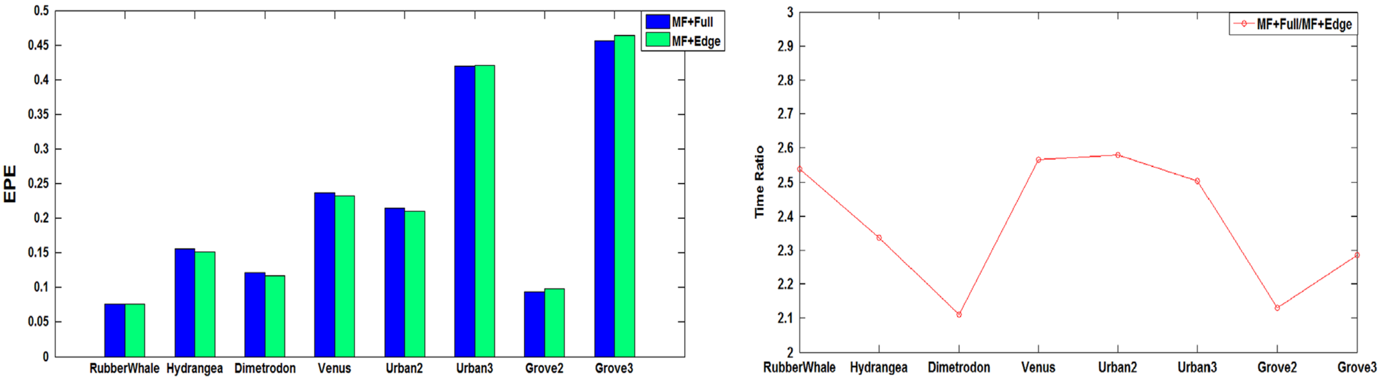


Fig. 1. Comparison of the EPE accuracy (left) and time consumption ratio (right) of "Classic+NL" algorithm with two different MF strategies – MF the Sobel detected edge parts and MF the full flow field on the Middlebury dataset sequences.

spatial averaging over a neighborhood around a point $M(i,j)$:

$$\begin{aligned} S(i,j;\sigma_s;\sigma_d) &= G(i,j;\sigma_s) * ((\nabla M(i,j;\sigma_d))(\nabla M(i,j;\sigma_d))^T) \\ &= G(i,j;\sigma_s) * \begin{bmatrix} M_x^2 & M_xM_y \\ M_xM_y & M_y^2 \end{bmatrix}_{|(i,j)} \\ &= G(i,j;\sigma_s) * S_0 = \begin{bmatrix} A & B \\ B & C \end{bmatrix}_{|(i,j)} \end{aligned} \tag{4}$$

where $S(i,j;\sigma_s;\sigma_d)$ is a symmetric, positive semi-definite matrix. The $*$ denotes the convolution operator. M_x and M_y are the derivatives of M in the x- and y-directions, respectively. They are calculated according to $M_x = \partial_x(G(i,j;\sigma_d) * M(i,j))$ and $M_y = \partial_y(G(i,j;\sigma_d) * M(i,j))$. $G(i,j;\sigma_d)$ is a Gaussian kernel with standard deviation σ_d (σ_d refers to the local scale). It is used to pre-smooth the data set before computing the derivatives. $S(i, j)$ is spatially smoothed according to a Gaussian kernel $G(i,j;\sigma_s)$ with standard deviation σ_s (σ_s refers to the integration scale) in a local neighborhood ($N \times N$). Due to spatial smoothing, noise is removed. Moreover, the neighboring structure information will be integrated into the center position (i, j). Since the two convolutions are linear operators, the ST $S(i,j;\sigma_s;\sigma_d)$ is referred to as a linear ST.

$S(i,j;\sigma_s;\sigma_d)$ can be considered as a covariance matrix of a two-dimensional distribution of gradient directions in a specified neighborhood of a point. Local structure information (e.g. orientation and magnitude) is presented in it. Its eigenvalues (λ_1, λ_2) are widely used to analyze the local structures of the data set $M(x, y)$.

The two eigenvalues λ_1 and λ_2 of the ST are non-negative, and can be computed by the following:

$$\lambda_{1,2} = \frac{(A+C) \pm \sqrt{(A-C)^2 + 4B^2}}{2} \tag{5}$$

The eigenvalues have different characteristics according to different local structures. For example, if a pixel is close to an edge, there should be a strong local orientation along the edge, and one eigenvalue will be large while the other one will be small.

3.1.2. 2D Harris edge detector

Based on the specific characteristics of the eigenvalues, some edge detectors are proposed to identify edges [28,29]. Due to the advantages of robustness to rotation, scale and noise, the Harris detector [29] is still one of the best methods to extract edges. Its response function is computed as follows:

$$H = \det(S) - k\,\mathrm{trace}^2(S) = \lambda_1\lambda_2 - k(\lambda_1+\lambda_2)^2 \tag{6}$$

The parameter k determines the extraction accuracy. The larger the k, the less sensitive the detector is to identify local structures. Empirically, k is set between 0.04 and 0.06. By checking the response value H of each point, edge point is identified if $H < 0$.

3.1.3. 3D spatial-scale Harris edge detector

Scale space theory [30] explains that scale is a parameter of the image resolution. Multi-scale images can be obtained by smoothing the original image with a series of Gaussian kernels $G(\sigma)$ with different standard deviations σ. Hence, the deviation σ is referred to scale. Ren [27] was the first to demonstrate that multi-scale processing significantly improves the image boundary detection. Liu et al. [28] stated that there is an optimal edge scale for each edge point. How to determine the optimal edge scale is difficult. In this work, similar as [28], we extend the spatial 2D edge detector (Eq. (6)) into scale space, and a spatial-scale 3D Harris edge detector is constructed. The spatial-scale edge detector should satisfy the following scale constraint:

$$\nabla M(x,y;\sigma_{d\Gamma}) = \frac{\partial |\,|\nabla M|\,|}{\partial \sigma_{d\Gamma}} = 0 \tag{7}$$

$M(x,y;\sigma_d)$ ($\sigma_d = \sigma_{d1}, \sigma_{d2}, \sigma_{d3}, \ldots$) is multi-scale and, it is generated by smoothing $M(x, y)$ with a series of local scale $\sigma_{d\Gamma}$ ($\Gamma = 1, 2, 3, \ldots$). $M(x,y;\sigma_{d\Gamma})_{|\sigma_{d\Gamma}}$ represents the first derivative of $M(x,y;\sigma_{d\Gamma})$. If $M(i, j)$ is an edge point, it should satisfy constraint Eq. (7). Transforming the zero-crossing $\nabla M(i,j;\sigma_{d\Gamma})$ to a local maximum form gives [28]:

$$\nabla \overline{M}_{\sigma_{d\Gamma}} = \mathrm{sign}(\nabla M_{\sigma_{d\Gamma}})\left(1 - \frac{\nabla M_{\sigma_{d\Gamma}}}{\max(\mathrm{abs}(\nabla M_{\sigma_{d\Gamma}}))}\right) \tag{8}$$

The spatial domain 2D ST can be updated to a spatial-scale 3D form using the scale information:

$$\begin{aligned} S(i,j;\sigma_s;\sigma_{d\Gamma}) &= G(i,j;\sigma_s) * ((\nabla M(i,j;\sigma_{d\Gamma}))(\nabla M(i,j;\sigma_{d\Gamma}))^T) \\ &= G(i,j;\sigma_s) * \begin{bmatrix} M_x^2 & M_xM_y & M_x\overline{M}_{\sigma_{d\Gamma}} \\ M_xM_y & M_y^2 & M_y\overline{M}_{\sigma_{d\Gamma}} \\ M_x\overline{M}_{\sigma_{d\Gamma}} & M_y\overline{M}_{\sigma_{d\Gamma}} & \overline{M}^2_{\sigma_{d\Gamma}} \end{bmatrix} \end{aligned} \tag{9}$$

Similar as the 3D spatial-time Harris interest point detector [31], a 3D spatial-scale Harris response function is constructed as follows:

$$H(\sigma_{d\Gamma}) = \det(S(\sigma_{d\Gamma})) - k\,\mathrm{trace}^3(S(\sigma_{d\Gamma})) = \lambda_1\lambda_2\lambda_3 - k(\lambda_1+\lambda_2+\lambda_3)^2 \tag{10}$$

where $H(\sigma_{d\Gamma})$ is the response function at each local scale $\sigma_{d\Gamma}$. It is the measure of edge saliency. λ_1, λ_2 and λ_3 measure the local variations in both spatial and scale space. We select $k=0.06$ in our experiments.

Edge detection: At every local scale $\sigma_{d\Gamma}$ ($\Gamma = 1, 2, 3, \ldots$), we can obtain a 3D Harris edge response $H(\sigma_{d\Gamma})$ of each point $M(i, j)$. Based on the characteristic that sharp variations occur along both scale and spatial directions, we propose the following criterion to determine edge points.

$$\begin{cases} H(\sigma_{d\Gamma}) < 0 \\ H_{|\sigma_{d\Gamma}} = 0 \;\;\&\;\; H_{|\sigma_{d\Gamma}\sigma_{d\Gamma}} < 0 \end{cases} \tag{11}$$

$H_{|\sigma_{d\Gamma}}$ is the first derivative of $H(\sigma_{d\Gamma})$ with respect to the local scale $\sigma_{d\Gamma}$, and $H_{|\sigma_{d\Gamma}\sigma_{d\Gamma}}$ is the second derivative of $H(\sigma_{d\Gamma})$ with respect to local scale $\sigma_{d\Gamma}$. This criterion ensures us to detect the strongest edge points among the scales. In our experiments, we set the multi-scales σ_d to [0, 0.5, 0.75, 1.0].

3.2. Nonlinear 3D spatial-scale Harris edge detection

Gaussian smoothing is simple and can be implemented efficiently. However, it will over-blur image details. This drawback can be overcome by considering both spatial and intensity similarities between pixels in averaging the weight design. A 2D GBF based nonlinear ST has been proposed for better corner detection [22]. We apply the GBF strategy to our 3D spatial-scale ST to improve the performances of edge detection.

The gradient intensity distance for a neighboring pixel $n(i',j') \in N(M(i,j))$ (N denotes the neighborhood) to the central pixel $M(i, j)$ is given by the following:

$$D_g(M(i,j), n(i',j')) = \sqrt{(M(i,j)_x - n(i',j')_x)^2 + (M(i,j)_y - n(i',j')_y)^2} \tag{12}$$

The spatial distance of a neighboring point $n(i',j') \in N(M(i,j))$ to the central point $M(i, j)$ is defined as follows:

$$D_s(M(i,j), n(i',j')) = \sqrt{(i-i')^2 + (j-j')^2} \tag{13}$$

A bilateral weighting function for each point is constructed as follows:

$$BF(\sigma_s;\sigma_g)_{|(i,j)} = \frac{W_s(D_s(M(i,j), n(i',j')))W_g(D_g(M(i,j), n(i',j')))}{\sum_{n(i',j') \in N(M(i,j))} W_s(D_s(M(i,j), n(i',j')); \sigma_s)W_g(D_g(M(i,j), n(i',j')); \sigma_g)} \tag{14}$$

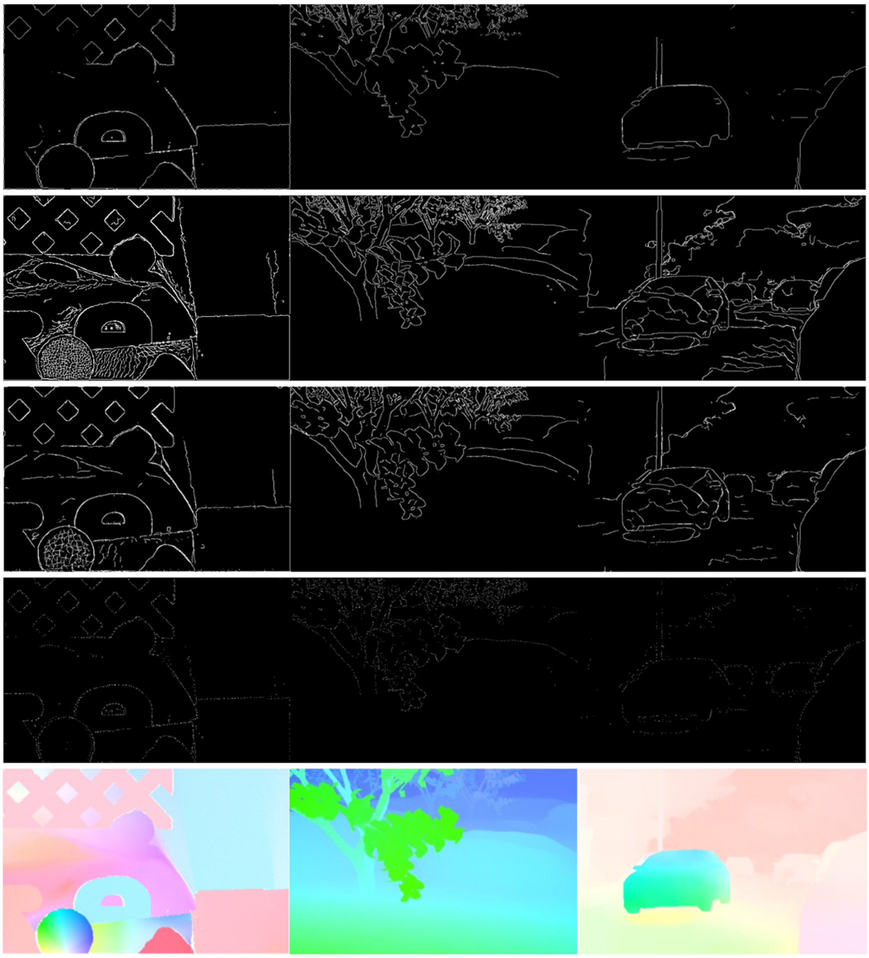


Fig. 2. From left to right: Detected edges of the ground truth (GT) flow fields of RubberWhale, Grove2 and Cameramotion (frames 49 and 50) sequences with different edge detectors. First line: detected edges with Sobel detector. Second line: detected edges with canny detector. Third line: detected edges with 2MM2 edge detector. Fourth line: detected edges with our nonlinear 3D Harris edge detector. Fifth line: GT flow fields of the three sequences.

where the spatial weight W_s and the gradient weight W_g are defined as follows:

$$W_s = \exp\left(-\frac{(D_s(M(i,j),n(i',j')))^2}{2\sigma_s^2}\right), \quad W_g = \exp\left(-\frac{(D_g(M(i,j),n(i',j')))^2}{2\sigma_g^2}\right) \tag{15}$$

A 3D GBF based nonlinear ST can be formed as follows:

$$\mathrm{BFS}(i,j;\sigma_s;\sigma_g;\sigma_d) = \mathrm{BF}(\sigma_s;\sigma_g)*((\nabla M(i,j;\sigma_d))(\nabla M(i,j;\sigma_d))^T)$$
$$= \begin{bmatrix} \mathrm{BF}(\sigma_s;\sigma_g)*M_x^2 & \mathrm{BF}(\sigma_s;\sigma_g)*M_xM_y & M_x\overline{M}_{\sigma_d} \\ \mathrm{BF}(\sigma_s;\sigma_g)*M_xM_y & \mathrm{BF}(\sigma_s;\sigma_g)*M_y^2 & M_y\overline{M}_{\sigma_d} \\ M_x\overline{M}_{\sigma_d} & M_y\overline{M}_{\sigma_d} & \overline{M}^2_{\sigma_d} \end{bmatrix} \tag{16}$$

Using the 3D nonlinear ST to replace the 3D linear ST $S(i,j;\sigma_s;\sigma_{d\Gamma})$ (Eq. (9)) of the spatial-scale Harris response function Eq. (10), a nonlinear spatial-scale Harris Edge detector is formed. Different from Liu et al. [28] who applied a Gaussian smoothing $G(i,j;\sigma_s)$ to all nine elements of the 3D ST, we use the GBF $G(i,j;\sigma_s)$ to smooth the four spatial gradient elements (Eq. (16)). With this improvement, the computational complexity is reduced and the average smoothing is easy to be implemented. Fig. 2 shows the high performance of our nonlinear 3D Harris edge detector, with the comparison of three well-known intensity suitable edge detectors – Sobel detector, Canny detector, and the advanced Second Moment Matrix (2MM2) detector [32]. The Sobel detector always neglects some salient edge points, while the Canny detector and the 2MM2 detector take some high texture points for edges. Contrarily, our nonlinear 3D spatial-scale Harris edge detector selects the correct edge points by comparing their saliency at different scales.

3.2.1. *Hybrid GBF and Gaussian filter smoothing (HGBGF) technique*

Applying the GBF to substitute the Gaussian filter is a good measure to treat the drawbacks of the traditional linear ST [22], however, the runtime drastically increases. To simultaneously reduce the time consumption while preserving discontinuities is a hard task. We propose a HGBGF approach to solve the low efficiency issue. The idea is inspired by Rashwan et al. [33], who used a spatio-temporal gradient method to classify an image into homogeneous and textured regions, and smooth image gradients which belong to different regions with different tensor voting parameters. In this work, we segment the spatial-scale ST elements into discontinuity regions and non-discontinuity regions by analyzing the SNR of their spatial-scale gradients. The GBF is used to smooth the discontinuity regions while the traditional Gaussian filter is employed to non-discontinuity regions. This HGBGF technique sharply reduces the computational time when compare to the GBF technique. As shown in Fig. 4, the HGBGF based spatial-scale 3D Harris edge detector is much more efficient than the GBF based spatial-scale 3D Harris edge detector, more than 40% time is saved.

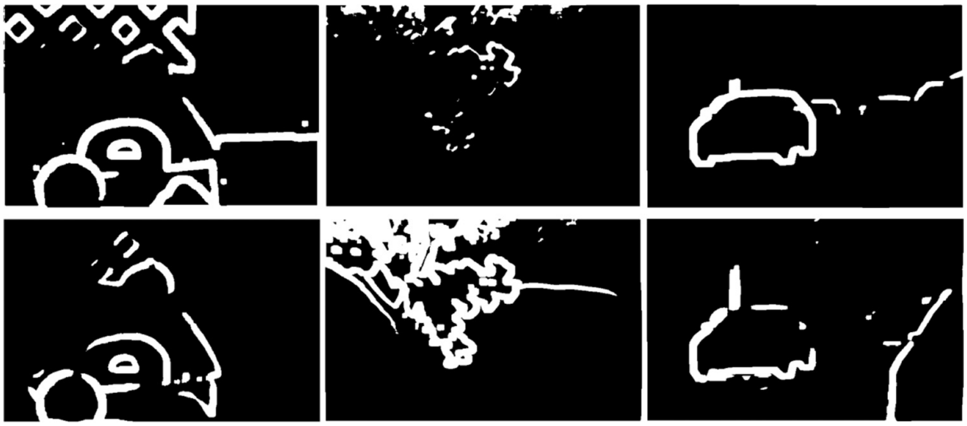


Fig. 3. From left to right: Classified discontinuity regions and non-discontinuity regions of the GT flow fields of RubberWhale, Grove2 and Cameramotion sequences with the SNR segmentation measure. First line: classified regions of the horizontal flow field u. Second line: classified regions of the vertical flow field v.

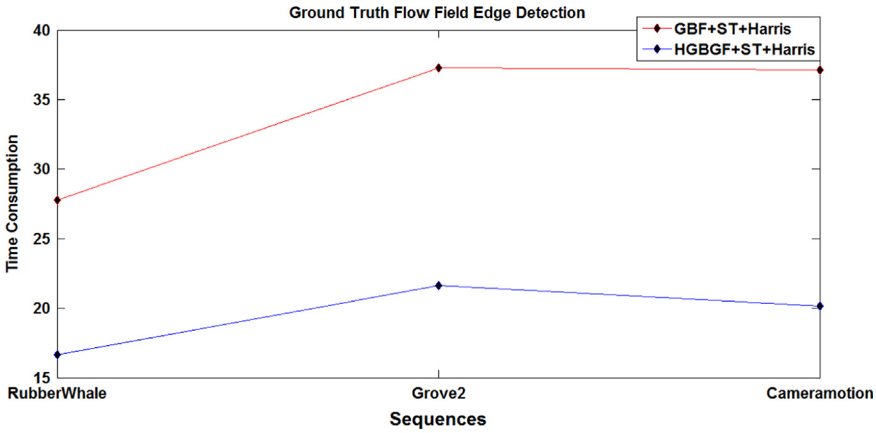


Fig. 4. Comparison of the time consumption of the HGBGF based and the GBF based 3D Harris edge detectors.

3.2.2. Spatial-scale gradient SNR segmentation measure

The spatial-scale gradient of each ST element can be calculated as follows:

$$||\nabla_3 M||_{(i,j)} = \sqrt{M_x^2 + M_y^2 + \overline{M}_{\sigma_{d\Gamma}}^2}\,|_{(i,j)} \tag{17}$$

Next, we calculate the mean μ and the standard deviation δ of each ST element $||\nabla_3 M||_{(i,j)}$ in a square window (in this work the window size is set to be 5×5). The SNR of each element is calculated as follows:

$$\mathrm{SNR} = 20 \cdot \log_{10}(\mu/\delta) \tag{18}$$

If an element is located at discontinuity, the change of the local scale $\sigma_{d\Gamma}$ causes its scale difference $\overline{M}_{\sigma_{d\Gamma}}$ to be large. Consequently, the standard deviation δ of its spatial-scale gradient is large, leading to a small SNR. In turn, if an element belongs to a non-discontinuity region, its spatial gradient (M_x, M_y) and scale difference $\overline{M}_{\sigma_{d\Gamma}}$ are small. Consequently, the standard deviation δ of its spatial-scale gradient is small, resulting in a large SNR. By setting a proper threshold τ to the SNR, the flow discontinuities can be well extracted (see Fig. 3). In our experiments, we set τ as follows:

$$\begin{cases} \tau = 36 + (i\mathrm{Scale} - 1) \times 9 \\ \tau = \min(\tau, \overline{\mathrm{SNR}}) \end{cases} \tag{19}$$

where the *i*Scale is the ordinal number of the selected multi-scales, min is the minimization operation, $\overline{\mathrm{SNR}}$ is the mean of SNR. As shown in Fig. 3, our SNR measure can accurately extract the ST elements which are located at discontinuities.

3.3. Piecewise occlusions detection

The combined flow divergence and pixel projection difference method is used to detect occlusions [18]:

$$Occ(x,y,t) = \exp\left(-\frac{(\mathrm{div}(x,y,t))^2}{2\sigma_1^2}\right) \cdot \exp\left(-\frac{(\mathrm{dif}(x,y,t))^2}{2\sigma_2^2}\right) \tag{20}$$

where $\mathrm{div}(x,y,t) = (\partial/\partial x)u(x,y,t) + (\partial/\partial y)v(x,y,t)$, $\mathrm{dif}(x,y,t) = I_1(x,y,t) - I_2(x+u, y+v, t+1)$, $\sigma_1 = 0.3$ and $\sigma_2 = 20$. The occlusion weight $Occ(x, y, t)$ indicates the occlusion status of each pixel. By experimentation (e.g. Fig. 5.), we find that: if $Occ(i, j)$ (at pixel (i, j)) is large, for example, it approximates 1, the pixel is non-occluded. If $Occ(i, j)$ is a little bit smaller, for example, it is smaller than 0.9, the pixel is not seriously occluded, it may be at the occlusion boundary or may be mistaken by its neighbors. A few of its neighbors are occluded. If $Occ(i, j)$ is much smaller than 1, for example, it is smaller than 0.6, the pixel is occluded, and most of its neighbors are occluded. If $Occ(i, j)$ is very small, for example, it smaller than 0.15, that means the pixel is seriously occluded. It is inside an occluded region, and its neighbors in a certain window are nearly all occluded. Based on this phenomenon, we classify the flow field vectors into 4 parts by piecewise thresholding $Occ(x, y, t)$:

$$Occ(x,y,t) = \begin{cases} NonOcc & (Occ(x,y,t) \geq 0.9) \\ Occ1 & (Occ(x,y,t) < 0.9)\ \&\ (Occ(x,y,t) \geq 0.6) \\ Occ2 & (Occ(x,y,t) < 0.6)\ \&\ (Occ(x,y,t) \geq 0.15) \\ Occ3 & (Occ(x,y,t) < 0.15) \end{cases} \tag{21}$$

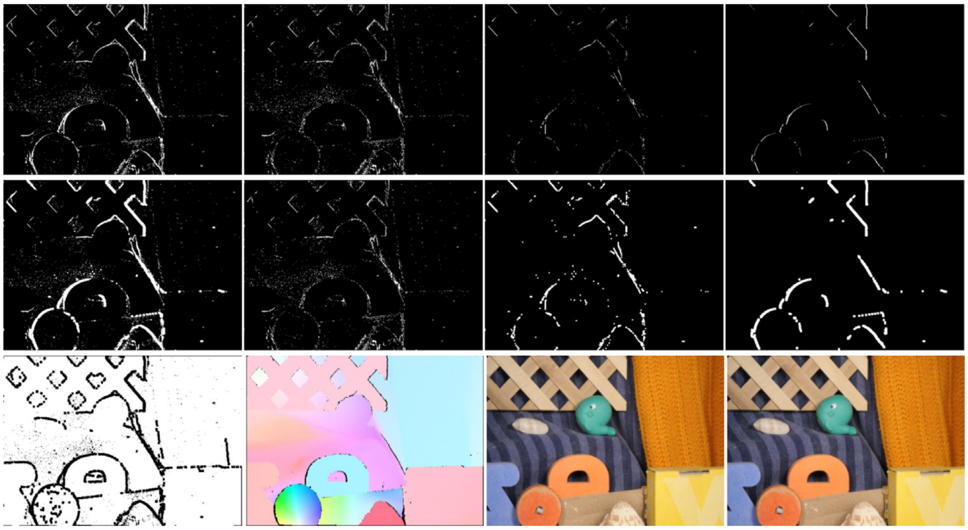


Fig. 5. Results of the piecewise threshold and piecewise threshold dilated occlusion detection method. First line (from left to right): *Occ*, *Occ*1, *Occ*2 and *Occ*3. Second line (from left to right): the dilated *Occ*, *Occ*1, *Occ*2 and *Occ*3. Third line (from left to right): non-occlusions *NonOcc* , the GT flow field of RubberWhale sequence, frame 10, and frame 11.

We dilate the three occlusion parts *Occ*1, *Occ*2 and *Occ*3, respectively, with three masks 1 × 1, 3 × 3 and 5 × 5 based on the characteristics we analyzed above. As shown in Fig. 5, the piecewise threshold dilated occlusion detection method can accurately identify the occlusions of the flow field.

4. Combined post-filtering

Post-filtering the estimated flow field is a good way to improve the accuracy. Not only some apparent outliers can be removed, some inaccurate flow components can also be corrected. For one estimated optical flow field, most of errors are distributed at the regions where the values have significant variation [32]. This is because the variational optical flow method is constrained to these assumptions: Lambertian surfaces with constant illumination conditions, no spatial discontinuities, and no object is allowed to be occluded. Hence, improving the performance at edge regions and occlusions is a suitable way to obtain better result. We propose a Combined Post-Filtering (CPF) method to handle this problem. With this method, four benefits are obtained: (1) over-smoothing is prevented and discontinuities are preserved, (2) by filtering the selected points rather than the full flow field, computational time is saved, (3) structure details are well preserved and flat regions blurring is avoided, and (4) false flow vectors at occlusions are further corrected using our BF instead of the MF.

4.1. Application of the WMF to edges

MF is a good way to remove outliers [14,16], it has the following properties: (1) choosing an appropriate filter window size ($N \times N$), and sorting the values of these points – in the window, in an increasing order. The value of point (i, j) is replaced by the middle value of the list as the filtered output, (2) there are as many larger as smaller values than the selected median value in the list (which means that the MF is constrained to be valid for a symmetrical list, hence, it is not suited to handle occlusions), and (3) the MF converges to a periodic solution if recursively executed on the flow field. Furthermore, when integrating the image and flow information to construct a weighted version (WMF), the discontinuities are better preserved. Different from [14] who simply applied the WMF to the Sobel extracted edges, we use the WMF to the edges which are identified with our nonlinear 3D spatial-scale Harris edge detector. Due to the better performance of our detector (see Fig. 2), the post-filtered result is more accurate than [14]. In practice, a 5 × 5 mask is used to dilate these detected edges before filtering.

4.2. Application of the BF to occlusions

MF [19] is effective for a symmetrical data set. Using MF to simply smooth the occlusions (which are severely asymmetrical), like [14], will generate some errors. For example, some points at occlusions which belong to foreground objects or surfaces would be wrongly replaced by values of the occluded objects or surfaces. Moreover, the estimated flow vectors are not accurate at occlusions. Because the pixels at the occlusions lack matching information – no correspondence is available in other frame, the variational method assigns the occluded pixels with certain displacement vectors by the diffusion operation. However, the current variational based diffusion lacks the occlusion handling mechanism and cannot discriminate the flow influence in the neighborhood of different regions very well. Therefore, we introduce a BF which can handle occlusions [17] to replace the MF to deal with these problems.

Xiao et al. [17] proposed a multi-cue driven adaptive BF to successfully smooth the optical flow field with highly desirable motion discontinuity preservation, moreover, the occlusions are properly handled. The average weight is calculated according to spatial proximity, image similarity, motion similarity, and occlusion status (Eq. (3)). The BF can distinguish incorrect flow vectors between different regions, and the flow vector of the occluded pixel (i, j) can be approximated by its neighboring non-occluded correct vectors, since the neighbors belong to the same surface or object as the occluded pixel (i, j). Of course, with this improvement, the smoothed result is better than the pure application of MF (see Fig. 6, Table 1). In our experiments, we set the half filter size of the BF to 9.

5. Experimental results and performance evaluation

In this section, the proposed algorithm, referred to as “Classic+CPF”, is compared to both classical and state-of-the-art optical flow algorithms. Both the proposed algorithm and the other

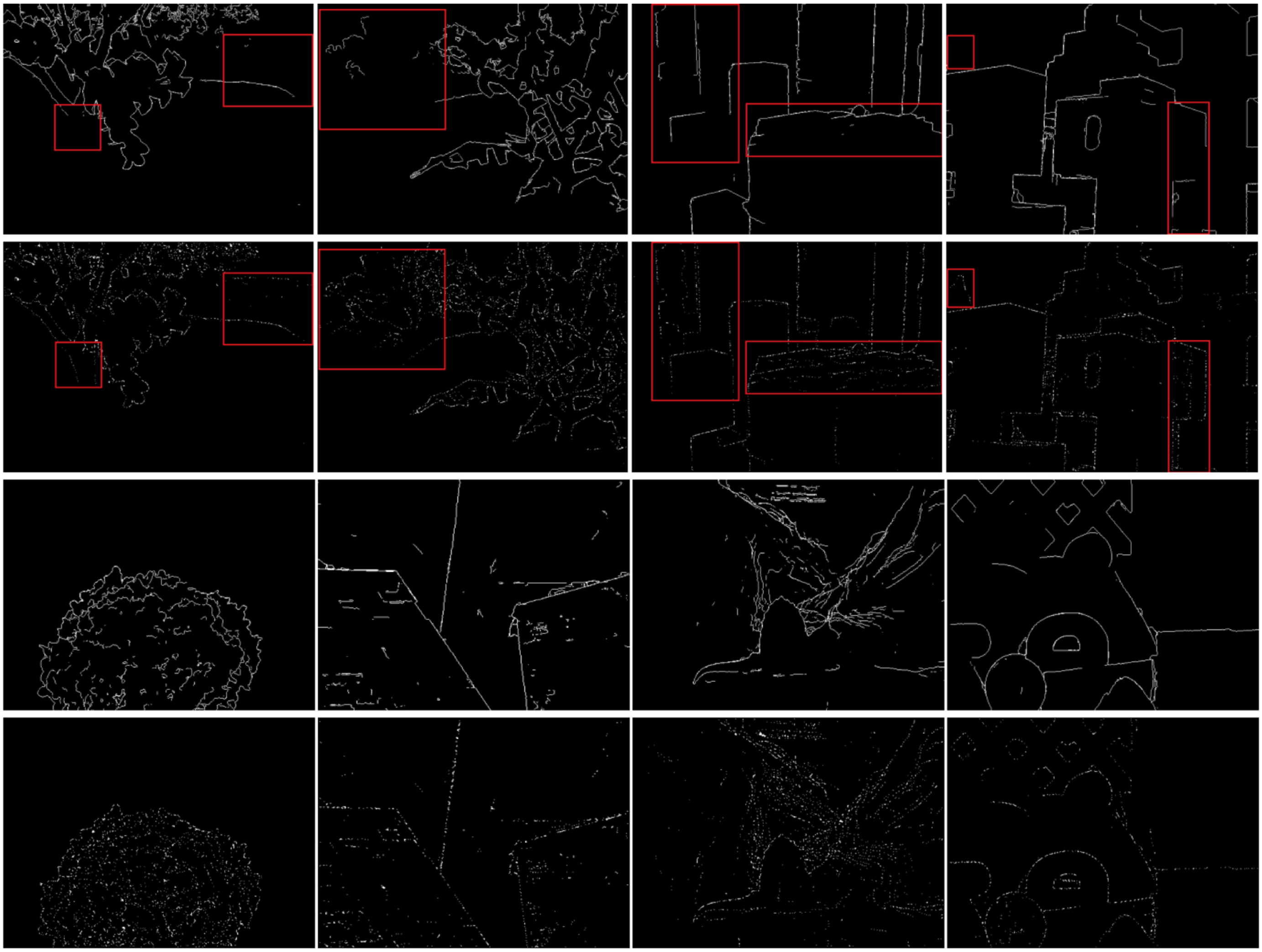


Fig. 6. Detected edges of the estimated flow fields of the "Classic+CPF" method. First line: detected edges of sequences Grove2, Grove3, Urban2 and Urban3 with the Sobel edge detector. Second line: detected edges of sequences Grove2, Grove3, Urban2 and Urban3 with the nonlinear 3D spatial-scale Harris edge detector. Third line: detected edges of sequences Venus, RubberWhale, Dimetrodon and Hydrangea with the Sobel edge detector. Fourth line: detected edges of sequences Venus, RubberWhale, Dimetrodon and Hydrangea with the nonlinear 3D spatial-scale Harris edge detector.

Table 1
Comparison of the proposed "Classic+CPF" method with two post-filtering variations: "Classic+WMF" and "Classic+BF".

Method	Urban2	Urban3	Grove2	Grove3	RubberWhale	Venus	Dimetrodon	Hydrangea
Classic+WMF	1.922/0.207	2.745/0.417	1.367/0.095	4.738/0.454	2.289/0.071	3.230/0.231	2.490/0.128	1.860/0.154
Classic+BF	2.105/0.262	3.794/0.507	1.508/0.104	5.415/0.529	2.281/0.071	3.318/0.233	2.507/0.127	1.821/0.153
Classic+CPF	**1.866/0.203**	**2.604/0.402**	**1.323/0.092**	**4.686/0.456**	**2.285/0.071**	**3.181/0.229**	**2.491/0.127**	**1.856/0.154**

algorithms [2,4,5,34] for comparison (their codes are available online) were implemented in MATLAB. Five most significant optical flow evaluation datasets are selected for testing: (1) the Middlebury dataset [2], (2) the UCL GT Optical Flow Dataset v1.2 [3], (3) the MIT motion annotation dataset [35], (4) the Football sequence [8], and (5) the newest KIT complex outdoor sequences dataset [36], which include both synthetic and real sequences. Two standard error measures – the Average Angular Error (AAE) and the average End Point Error (EPE) are applied to evaluate the accuracy. All the experiments are performed on a PC with an Intel Core i5-2410M 2.30 GHz processor and 4 GB memory. For coarse-to-fine estimation, we execute 3 warping iterations when the pyramid level is greater than 1, while 5 warping iterations when the pyramid level equals 1, to save computational time (the 10 warping iterations scheme at each pyramid level of the "Classic+NL" algorithm is time consuming). Many more pictures and experimental results are available at: http://www.projects.science.uu.nl/opticalflow/.

5.1. Effectiveness of the CPF method

The purpose of the first experiment is to verify the effectiveness of our CPF method – whether it can both remove outliers and handle occlusions. Eight synthetic test sequences from the benchmark Middlebury dataset [2] are used for evaluation. Table 1 shows the AAE and EPE of the "Classic+CPF" algorithm and its two related variations: (1) "Classic+WMF", which purely uses the WMF method to post-smooth the detected flow edges and occlusions and (2) "Classic+BF", which purely uses the BF method to post-smooth the detected flow edges and occlusions. As can be

Table 2
Comparison of the proposed "Classic+CPF" method to the methods "Classic+NL" [14],"Classic+NL-full" and "Classic+CPF (Sobel)" in terms of AAE/EPE/time on all eight test sequences from the benchmark Middlebury dataset [2].

Method	Urban2	Urban3	Grove2	Grove3
Classic+NL	2.058/0.216/8.050	2.590/0.379/7.920	1.491/0.103/10.30	5.070/0.476/9.560
Classic+NL-full	1.973/0.212/22.18	2.645/0.397/23.02	1.393/0.097/24.69	4.864/0.467/22.80
Classic+CPF(Sobel)	1.873/0.203/7.330	2.609/0.410/7.430	1.333/0.093/8.501	4.777/0.463/10.01
Classic+CPF	1.866/0.201/10.25	2.478/0.399/11.50	1.323/0.092/11.05	4.686/0.456/11.75
	Hydrangea	Venus	RubberWhale	Dimetrodon
Classic+NL	1.831/0.151/7.130	3.316/0.236/5.450	2.356/0.073/6.220	2.572/0.131/8.050
Classic+NL-full	1.902/0.156/17.02	3.435/0.249/11.65	2.378/0.074/16.79	2.624/0.135/17.34
Classic+CPF(Sobel)	1.877/0.156/5.390	3.211/0.229/3.780	2.277/0.071/4.750	2.485/0.128/5.760
Classic+CPF	1.856/0.154/8.750	3.181/0.229/6.020	2.285/0.071/9.050	2.491/0.127/9.200

seen, the proposed "Classic+CPF" algorithm attains the best performance. Table 1 demonstrates that the CPF method really integrates both the advantages of the WMF which can remove outliers while preserving the edges, and the advantages of the BF which can tackle occlusions while preserving discontinuities. For the sequence which includes lots of occlusions and motion discontinuities, such as the Urban3, Grove3 and Venus, our CPF method is even more effective. Specially, for the Hydrangea sequence, there is no large displacement, no illumination changes, few noise and high resolution, and plays a whole rotation movement. Hence, errors in the estimated flow field are primarily due to mismatching (one kind of occlusion [17]). The lowest AAE/EPE of the "Classic+BF" method also illustrates that using the BF to replace the WMF for smoothing occlusions is beneficial and necessary. In Table 2, we compare the accuracy (AAE and EPE) and efficiency (time) of four methods: (1) the baseline method "Classic+NL", (2) the method "Classic+NL-full", which uses the WMF to smooth the full flow field, (3) the method "Classic+CPF (Sobel)", which uses the Sobel edge detector instead of our nonlinear 3D spatial-scale Harris edge detector to extract edges, and (4) the proposed method "Classic+CPF". By comparing results of the method "Classic+NL" and the method "Classic+NL-full", we can find that for most of the sequences, like Urban3, Venus, RubberWhale, Dimetrodon and Hydrangea, the method "Classic+NL" is more accurate. Most importantly, the efficiency of the method "Classic+NL" is much higher (at least 2 time faster, see Fig. 1 for more detail) than the method "Classic+NL-full". Therefore, post-smoothing the whole flow field is no useful. By comparing results of the method "Classic+CPF (Sobel)" and the method "Classic+CPF", we can see that the EPE of all the 8 sequences of the method "Classic+CPF" is equal or smaller than the method "Classic+CPF (Sobel)". The AAE of the method "Classic+CPF" is also more accurate, except for the RubberWhale sequence and the Dimetrodon sequence. But the AAE difference of the two sequences between the two methods is nearly the same (less than 0.5% difference). This comparison illustrates that our proposed nonlinear 3D spatial-scale Harris edge detector is effective. Fig. 6 shows the detected edges of the 8 test sequences by the Sobel edge detector and our edge detector. The red rectangle highlighted regions of the 4 sequences can explain why the "Classic+CPF" method performs better than the "Classic+CPF (Sobel)" method: our edge detector extracts nearly all the significant motion boundaries, while the Sobel edge detector misses many important motion boundaries. For the other four sequences (e.g. Venus, RubberWhale, Dimetrodon and Hydrangea) which without highlights, the detected edges by the Sobel detector and the nonlinear 3D spatial-scale Harris detector are similar, their AEE and EPE between the two methods are approximately the same. Most importantly, the proposed "Classic+CPF" method obtains much more accurate results than the "Classic+NL" method, especially if the sequence contains serious occlusions. Such as the Urban2, Grove3 and Venus, the AAE/EPE is reduced from 2.058/0.216 to 1.866/0.201, from 5.070/0.476 to 4.686/0.456, and from 3.316/0.236 to 3.181/0.229, respectively. The accuracy improvement is about 5% or more. For the Hydrangea sequence, the result of our "Classic+CPF" method is lightly worse than the "Classic+NL" method. That is because the Hydrangea contains only a few outliers but abundant of small structures, our nonlinear 3D spatial-scale Harris detector extracts nearly all its small structural edges, which results in over-smoothing its some textures and details with the filters. In contrast, the over-smoothing is reduced in "Classic+NL" method as the Sobel detector misses some structural edges (see Fig. 6). Comparing the AAE/EPE of the "Classic+NL" method and the "Classic+NL" method, we can also find this conclusion. Therefore, our nonlinear 3D spatial-scale Harris edge detector based CPF method as well as some other post-smoothing variational methods are not suitable for the sequence which is of high quality, high resolution, contains no large displacement, no serious occlusions and few outliers, but contains plenty of small structures. Thus, for this kind of sequences, we suggest to post-smooth only some distinctive motion boundaries, and the detector which extracts few but significant edges should be used.

To further certify the effectiveness of the CPF method, we test our "Classic+CPF" method on two other well-known optical flow evaluation datasets [35,3], which contains different complex conditions with [2]. The sequences of [35] are reconstructed in different way with [3]. The supplied GT of [35,3] can be used to quantitative analyze the performance of our method. Tables 3 and 4 show the AAE/EPE of all the test sequences of [35], and the majority of test sequences of [3] (the selected sequences can represent all the sequences on [3]). As can be seen from Table 3, significant reductions of the AAE and EPE are achieved nearly for all sequences, except the Toy. The Toy sequence contains similar conditions as the Hydrangea, which explains why the CPF method is not beneficial at this moment. In Table 4, except the Crates1Htxtr2, all the other 15 sequences obtain a more accurate flow field. This is a noticeable problem which needs us to deeply study. Since we just set the half filter size of the BF fixed to 9, is not adaptive. However, like the Crates1Htxtr2 sequence, it contains large displacements which are more than 50 pixels, hence the small filtering window size of the BF is not valid. To find an efficient adaptive method to set the filtering window size is a promising way to improve the performance of the CPF method.

Fig. 7 shows the visual flow fields of 4 challenging sequences – *Urban3*, *Venus*, *drop9Txtr2* and *Fish*. The *Urban3* sequence contains serious occlusions, small scale moving part and multiple motion, thus, they cannot be estimated correctly in the baseline method [14]. In contrast, our method greatly reduces the errors caused by occlusions, and the shape of the small scale roof at the top left corner is successfully recovered. Comparing the

Table 3
Comparison of the proposed “Classic+CPF” method to the “Classic+NL” method [14] in terms of AAE/EPE on all five sequences from the MIT dataset [35].

Method	Fish	Cameramotion	Table	Hand	Toy
Classic+NL	19.988/0.768	6.428/0.572	3.862/1.198	16.228/1.838	**2.548/0.562**
Classic+CPF	15.306/0.627	5.785/0.533	3.750/1.140	14.728/1.684	2.621/0.579

Table 4
Comparison of the proposed “Classic+CPF” method to the “Classic+NL” method [14] in terms of AAE/EPE on some sequences from the UCL Dataset v1.2 [3].

Method	YoesmiteSun	GroveSun	Crates1	Robot	Sponza1	Crates1Htxtr2	Brickbox1t1	GrassSky9
Classic+NL	3.062/0.157	5.756/0.270	5.006/3.435	8.566/1.471	12.557/1.345	**2.339/0.382**	0.708/0.239	0.836/0.319
Classic+CPF	2.705/0.148	5.313/0.248	4.439/3.286	6.225/1.150	11.892/1.274	3.652/1.596	0.618/0.237	0.726/0.306
	TxtRMovemet	blow1Txtr1	blow19Txtr2	drop1Txtr1	drop9Txtr2	roll1Txtr1	roll9Txtr2	street1Txtr1
Classic+NL	0.120/0.101	0.535/0.025	1.956/0.187	1.246/0.042	5.240/1.755	0.089/0.002	0.420/0.012	3.464/3.961
Classic+CPF	0.100/0.097	0.428/0.023	1.609/0.192	1.010/0.041	4.706/2.247	0.076/0.001	0.360/0.011	3.242/4.045

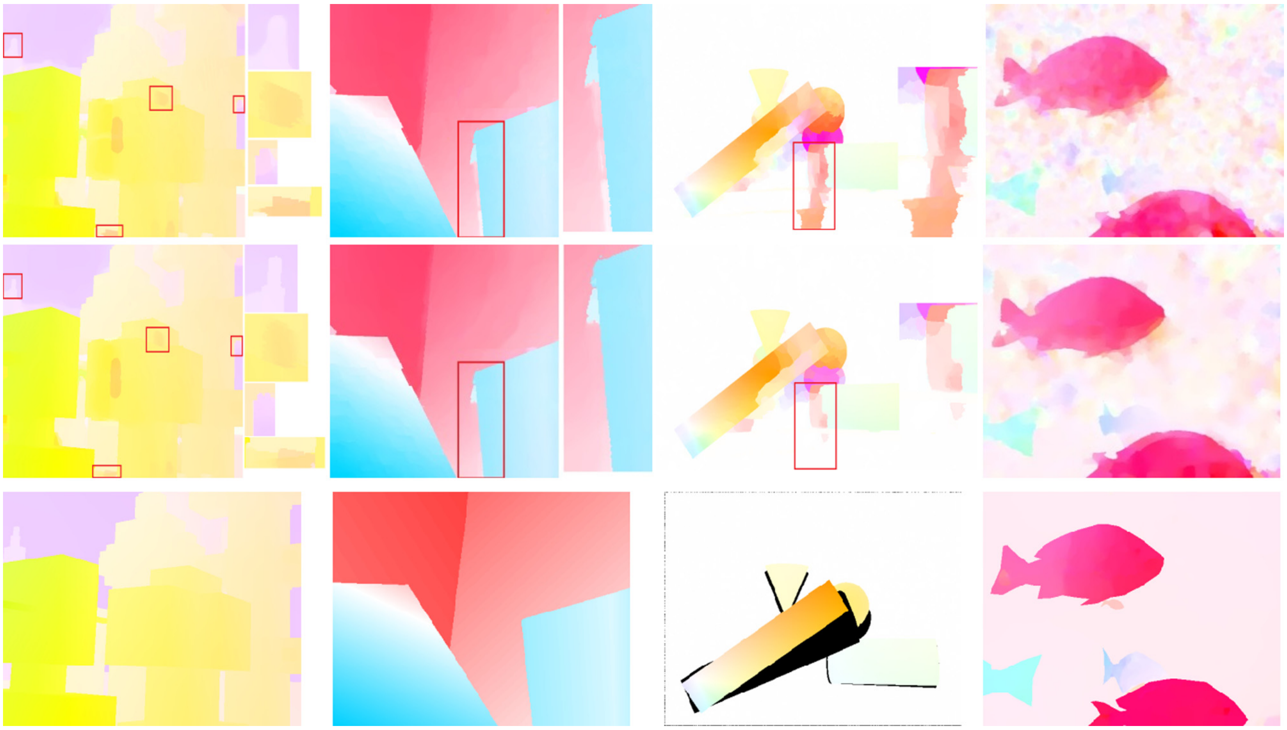


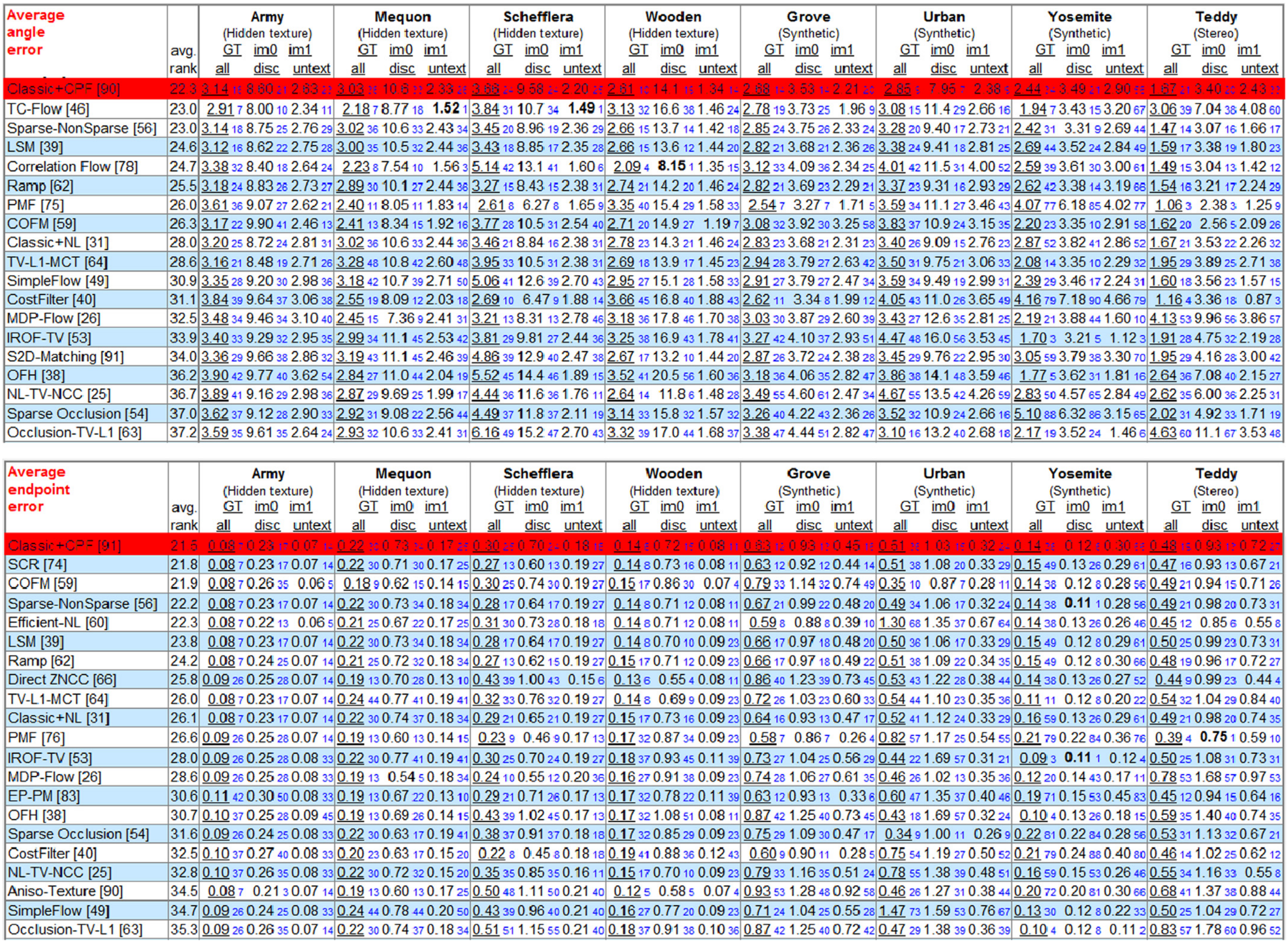
Fig. 7. Comparison of the estimated flow fields (each column, from left to righ) of the proposed “Classic+CPF” algorithm to the “Classic+NL” algorithm [14] on sequences Urban3, Venus, Fish (frames 145 and 146) and drop9Txtr2. First line: estimated flow field with the “Classic+NL” algorithm. Second line: estimated flow field with our “Classic+ CPF” algorithm. Third line, GT of these sequences.

highlighted regions of the *Venus* sequence and the *drop9Txtr2* sequence, we can see that the CPF method has good performance to handle occlusions: the lost motion in [14] is correctly estimated and the mismatching errors are significantly reduced. For the *Fish* sequence, the computed flow field of [14] contains excessive noise and outliers, by contrast, due to the contribution of the CPF method, the noise and outliers are excellently smoothed out and the motion boundaries are well preserved.

Fig. 8 shows the evaluation results on the Middlebury optical flow benchmark [2]. Both the AAE and EPE are ranked 17th among 91 methods. Most importantly, our method outperforms nearly all the TV-L1 based non-local (NL) methods, such as “Efficient-NL”, “Classic+NL”, “NL-TV-NCC” and “Occlusion-TV-L1” in the rank list. To be more precise, comparing our “Classic+CPF” algorithm with the “Classic+NL” algorithm to the “Urban” sequence, not only the accuracy is improved (from 3.40/0.52 to 2.85/0.51) the consumption time is greatly reduced (640–972 s), which is nearly 35%.

5.2. Universality of the CPF method

In this experiment, we want to demonstrate that the propose CPF method is useful for almost all the variational optical flow methods. Tables 5 and 6 show that when integrating our CPF method into four classical and state-of-the-art optical flow algorithms (e.g. HS [4], BA [6], DN [7], CP [34]), their performances are further improved. And the CPF method is especially effective for the noisy and serious occluded sequences. As shown in Tables 5 and 6, for the noisy fish and Cameramotion sequences, and the severely occluded drop9Txtr2 sequence, the results of the original HS [4] algorithm, BA [6] algorithm, DN [7] algorithm and CP [34] algorithm are quite inaccurate. The proposed CPF method smoothes the intermediate flow field during incremental estimation and warping, outliers are removed and occlusions are partially handled, hence, the flow accuracy is improved. Take the BA [6] algorithm for example, the AAE of the fish and the drop9Txtr2 is



Average angle error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Classic+CPF [90]	22.3	3.14 18	8.60 21	2.63 22	3.03 38	10.6 33	2.33 28	3.66 24	9.58 24	2.20 25	2.61 10	14.1 19	1.34 14	2.68 14	3.53 14	2.21 20	2.85 9	7.95 7	2.38 5	2.44 34	3.49 21	2.90 55	1.67 21	3.40 20	2.43 33
TC-Flow [46]	23.0	2.91 7	8.00 10	2.34 11	2.18 7	8.77 18	**1.52** 1	3.84 31	10.7 34	**1.49** 1	3.13 32	16.6 38	1.46 24	2.78 19	3.73 25	1.96 9	3.08 15	11.4 29	2.66 16	1.94 7	3.43 15	3.20 67	3.06 39	7.04 38	4.08 60
Sparse-NonSparse [56]	23.0	3.14 18	8.75 25	2.76 29	3.02 36	10.6 33	2.43 34	3.45 20	8.96 19	2.36 29	2.66 15	13.7 14	1.42 18	2.85 24	3.75 26	2.33 24	3.28 20	9.40 17	2.73 21	2.42 31	3.31 9	2.69 44	1.47 14	3.07 16	1.66 17
LSM [39]	24.6	3.12 16	8.62 22	2.75 28	3.00 35	10.5 32	2.44 36	3.43 18	8.85 17	2.35 28	2.66 15	13.6 12	1.44 20	2.82 21	3.68 21	2.36 26	3.38 24	9.41 18	2.81 25	2.69 44	3.52 24	2.84 49	1.59 17	3.38 19	1.80 23
Correlation Flow [78]	24.7	3.38 32	8.40 18	2.64 24	2.23 8	7.54 10	1.56 3	5.14 42	13.1 41	1.60 6	2.09 4	**8.15** 1	1.35 15	3.12 33	4.09 36	2.34 25	4.01 42	11.5 31	4.00 52	2.59 39	3.61 30	3.00 61	1.49 15	3.04 13	1.42 12
Ramp [62]	25.5	3.18 24	8.83 26	2.73 27	2.89 30	10.1 27	2.44 36	3.27 15	8.43 15	2.38 31	2.74 21	14.2 20	1.46 24	2.82 21	3.69 23	2.29 21	3.37 23	9.31 16	2.93 29	2.62 42	3.38 14	3.19 66	1.54 16	3.21 17	2.24 29
PMF [75]	26.0	3.61 36	9.07 27	2.62 21	2.40 11	8.05 11	1.83 14	2.61 8	6.27 8	1.65 9	3.35 40	15.4 29	1.58 33	2.54 7	3.27 7	1.71 5	3.59 34	11.1 27	3.46 43	4.07 77	6.18 85	4.02 77	1.06 3	2.38 3	1.25 9
COFM [59]	26.3	3.17 22	9.90 41	2.46 13	2.41 13	8.34 15	1.92 16	3.77 28	10.5 31	2.54 40	2.71 20	14.9 27	1.19 7	3.08 32	3.92 30	3.25 58	3.83 37	10.9 24	3.15 35	2.20 23	3.35 10	2.91 58	1.62 20	2.56 5	2.09 26
Classic+NL [31]	28.0	3.20 25	8.72 24	2.81 31	3.02 36	10.6 33	2.44 36	3.46 21	8.84 16	2.38 31	2.78 23	14.3 21	1.46 24	2.83 23	3.68 21	2.31 23	3.40 26	9.09 15	2.76 23	2.87 52	3.82 41	2.86 52	1.67 21	3.53 22	2.26 32
TV-L1-MCT [64]	28.6	3.16 21	8.48 19	2.71 26	3.28 48	10.8 42	2.60 48	3.95 33	10.5 31	2.38 31	2.69 18	13.9 17	1.45 23	2.94 28	3.79 27	2.63 42	3.50 31	9.75 21	3.06 33	2.08 14	3.35 10	2.29 32	1.95 29	3.89 25	2.71 38
SimpleFlow [49]	30.9	3.35 28	9.20 30	2.98 36	3.18 42	10.7 39	2.71 50	5.06 41	12.6 39	2.70 43	2.95 27	15.1 28	1.58 33	2.91 27	3.79 27	2.47 34	3.59 34	9.49 19	2.99 31	2.39 29	3.46 17	2.24 31	1.60 18	3.56 23	1.57 15
CostFilter [40]	31.1	3.84 39	9.64 37	3.06 38	2.55 19	8.09 12	2.03 18	2.69 10	6.47 9	1.88 14	3.66 45	16.8 40	1.88 43	2.62 11	3.34 8	1.99 12	4.05 43	11.0 26	3.65 49	4.16 79	7.18 90	4.66 79	1.16 4	3.36 18	0.87 3
MDP-Flow [26]	32.5	3.48 34	9.46 34	3.10 40	2.45 15	7.36 9	2.41 31	3.21 13	8.31 13	2.78 46	3.18 36	17.8 46	1.70 38	3.03 30	3.87 29	2.60 39	3.43 27	12.6 35	2.81 25	2.19 21	3.88 44	1.60 10	4.13 53	9.96 56	3.86 57
IROF-TV [53]	33.9	3.40 33	9.29 32	2.95 35	2.99 34	11.1 45	2.53 42	3.81 29	9.81 27	2.44 36	3.25 38	16.9 43	1.78 41	3.27 42	4.10 37	2.93 51	4.47 48	16.0 56	3.53 45	1.70 3	3.21 5	1.12 3	1.91 28	4.75 32	2.19 28
S2D-Matching [91]	34.0	3.36 29	9.66 38	2.86 32	3.19 43	11.1 45	2.46 39	4.86 39	12.9 40	2.47 38	2.67 17	13.2 10	1.44 20	2.87 26	3.72 24	2.38 28	3.45 29	9.76 22	2.95 30	3.05 59	3.79 38	3.30 70	1.95 29	4.16 28	3.00 42
OFH [38]	36.2	3.90 42	9.77 40	3.62 54	2.84 27	11.0 44	2.04 19	5.52 45	14.4 46	1.89 15	3.52 41	20.5 56	1.60 36	3.18 36	4.06 35	2.82 47	3.86 38	14.1 48	3.59 46	1.77 5	3.62 31	1.81 16	2.64 36	7.08 40	2.15 27
NL-TV-NCC [25]	36.7	3.89 41	9.16 29	2.98 36	2.87 29	9.69 25	1.99 17	4.44 36	11.6 36	1.76 11	2.64 14	11.8 6	1.48 28	3.49 55	4.60 61	2.47 34	4.67 55	13.5 42	4.26 59	2.83 50	4.57 65	2.84 49	2.62 35	6.00 36	2.25 31
Sparse Occlusion [54]	37.0	3.62 37	9.12 28	2.90 33	2.92 31	9.08 22	2.56 44	4.49 37	11.8 37	2.11 19	3.14 33	15.8 32	1.57 32	3.26 40	4.22 43	2.36 26	3.52 32	10.9 24	2.66 16	5.10 88	6.32 86	3.15 65	2.02 31	4.92 33	1.71 19
Occlusion-TV-L1 [63]	37.2	3.59 35	9.61 35	2.64 24	2.93 32	10.6 33	2.41 31	6.16 49	15.2 47	2.70 43	3.32 39	17.0 44	1.68 37	3.38 47	4.44 51	2.82 47	3.10 16	13.2 40	2.68 18	2.17 19	3.52 24	1.46 6	4.63 60	11.1 67	3.53 48

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
Classic+CPF [91]	21.5	0.08 7	0.23 17	0.07 14	0.22 30	0.73 34	0.17 25	0.30 25	0.70 24	0.18 18	0.14 8	0.72 15	0.08 11	0.63 12	0.93 13	0.45 15	0.51 38	1.03 15	0.32 24	0.14 38	0.12 8	0.30 66	0.48 19	0.93 13	0.72 27
SCR [74]	21.8	0.08 7	0.23 17	0.07 14	0.22 30	0.71 30	0.17 25	0.27 13	0.60 13	0.19 27	0.14 8	0.73 16	0.08 11	0.63 12	0.92 12	0.44 14	0.51 38	1.08 20	0.33 29	0.15 49	0.13 26	0.29 61	0.47 16	0.93 13	0.67 21
COFM [59]	21.9	0.08 7	0.26 35	0.06 5	0.18 9	0.62 15	0.14 15	0.30 25	0.74 30	0.19 27	0.15 17	0.86 30	0.07 4	0.79 33	1.14 32	0.74 49	0.35 10	0.87 7	0.28 11	0.14 38	0.12 8	0.28 56	0.49 21	0.94 15	0.71 26
Sparse-NonSparse [56]	22.2	0.08 7	0.23 17	0.07 14	0.22 30	0.73 34	0.18 34	0.28 17	0.64 17	0.19 27	0.14 8	0.71 12	0.08 11	0.67 21	0.99 22	0.48 20	0.49 34	1.06 17	0.32 24	0.14 38	**0.11** 1	0.28 56	0.49 21	0.98 20	0.73 31
Efficient-NL [60]	22.3	0.08 7	0.22 13	0.06 5	0.21 25	0.67 22	0.17 25	0.31 30	0.73 28	0.18 18	0.14 8	0.71 12	0.08 11	0.59 8	0.88 8	0.39 10	1.30 68	1.35 37	0.67 64	0.14 38	0.13 26	0.26 46	0.45 12	0.85 6	0.55 8
LSM [39]	23.8	0.08 7	0.23 17	0.07 14	0.22 30	0.73 34	0.18 34	0.28 17	0.64 17	0.19 27	0.14 8	0.70 10	0.09 23	0.66 17	0.97 18	0.48 20	0.50 36	1.06 17	0.33 29	0.15 49	0.12 8	0.29 61	0.50 25	0.99 23	0.73 31
Ramp [62]	24.2	0.08 7	0.24 25	0.07 14	0.21 25	0.72 32	0.18 34	0.27 13	0.62 15	0.19 27	0.15 17	0.71 12	0.09 23	0.66 17	0.97 18	0.49 22	0.51 38	1.09 22	0.34 35	0.15 49	0.12 8	0.30 66	0.48 19	0.96 17	0.72 27
Direct ZNCC [66]	25.8	0.09 26	0.25 28	0.07 14	0.19 13	0.70 28	0.13 10	0.43 39	1.00 43	0.15 6	0.13 6	0.55 4	0.08 11	0.86 40	1.23 39	0.73 45	0.53 43	1.22 28	0.38 44	0.14 38	0.13 26	0.27 52	0.44 9	0.99 23	0.44 4
TV-L1-MCT [64]	26.0	0.08 7	0.23 17	0.07 14	0.24 44	0.77 41	0.19 41	0.32 33	0.76 32	0.19 27	0.14 8	0.69 9	0.09 23	0.72 26	1.03 23	0.60 33	0.54 44	1.10 23	0.35 36	0.11 11	0.12 8	0.20 22	0.54 32	1.04 29	0.84 40
Classic+NL [31]	26.1	0.08 7	0.23 17	0.07 14	0.22 30	0.74 37	0.18 34	0.29 21	0.65 21	0.19 27	0.15 17	0.73 16	0.09 23	0.64 16	0.93 13	0.47 17	0.52 41	1.12 24	0.33 29	0.16 59	0.13 26	0.29 61	0.49 21	0.98 20	0.74 35
PMF [76]	26.6	0.09 26	0.25 28	0.07 14	0.19 13	0.60 13	0.14 15	0.23 9	0.46 9	0.17 13	0.17 32	0.87 34	0.09 23	0.58 7	0.86 7	0.26 4	0.82 57	1.17 25	0.54 55	0.21 79	0.22 84	0.36 76	0.39 4	**0.75** 1	0.59 10
IROF-TV [53]	28.0	0.09 26	0.25 28	0.08 33	0.22 30	0.77 41	0.19 41	0.30 25	0.70 24	0.19 27	0.18 37	0.93 45	0.11 39	0.73 27	1.04 25	0.56 29	0.44 22	1.69 57	0.31 21	0.09 3	**0.11** 1	0.12 4	0.50 25	1.08 31	0.73 31
MDP-Flow [26]	28.6	0.09 26	0.25 28	0.08 33	0.19 13	0.54 5	0.18 34	0.24 10	0.55 12	0.20 36	0.16 27	0.91 38	0.09 23	0.74 28	1.06 27	0.61 35	0.46 26	1.02 13	0.35 36	0.12 20	0.14 43	0.17 11	0.78 53	1.68 57	0.97 53
EP-PM [83]	30.6	0.11 42	0.30 50	0.08 33	0.19 13	0.67 22	0.13 10	0.29 21	0.71 26	0.17 13	0.17 32	0.78 22	0.11 39	0.63 12	0.93 13	0.33 6	0.60 47	1.35 37	0.40 46	0.19 71	0.15 53	0.45 83	0.45 12	0.94 15	0.64 16
OFH [38]	30.7	0.10 37	0.25 28	0.09 45	0.19 13	0.69 26	0.14 15	0.43 39	1.02 45	0.17 13	0.17 32	1.08 51	0.08 11	0.87 42	1.25 40	0.73 45	0.43 18	1.69 57	0.32 24	0.10 4	0.13 26	0.18 15	0.59 35	1.40 40	0.74 35
Sparse Occlusion [54]	31.6	0.09 26	0.24 25	0.08 33	0.22 30	0.63 17	0.19 41	0.38 37	0.91 37	0.18 18	0.17 32	0.85 29	0.09 23	0.75 29	1.09 30	0.47 17	0.34 9	1.00 11	0.26 9	0.22 81	0.22 84	0.28 56	0.53 31	1.13 32	0.67 21
CostFilter [40]	32.5	0.10 37	0.27 40	0.08 33	0.20 23	0.63 17	0.15 20	0.22 8	0.45 8	0.18 18	0.19 41	0.88 36	0.12 43	0.60 9	0.90 11	0.28 5	0.75 54	1.19 27	0.50 52	0.21 79	0.24 88	0.40 80	0.46 14	1.02 25	0.62 12
NL-TV-NCC [25]	32.8	0.10 37	0.26 35	0.08 33	0.22 30	0.72 32	0.15 20	0.35 35	0.85 35	0.16 11	0.15 17	0.70 10	0.09 23	0.79 33	1.16 35	0.51 24	0.78 55	1.38 39	0.48 51	0.16 59	0.15 53	0.26 46	0.55 34	1.16 33	0.55 8
Aniso-Texture [90]	34.5	0.08 7	0.21 3	0.07 14	0.19 13	0.60 13	0.17 25	0.50 48	1.11 50	0.21 40	0.12 5	0.58 5	0.07 4	0.93 53	1.28 48	0.92 58	0.46 26	1.27 31	0.38 44	0.20 72	0.20 81	0.30 66	0.68 41	1.37 38	0.88 44
SimpleFlow [49]	34.7	0.09 26	0.24 25	0.08 33	0.24 44	0.78 44	0.20 50	0.43 39	0.96 40	0.21 40	0.16 27	0.77 20	0.09 23	0.71 24	1.04 25	0.55 28	1.47 73	1.59 53	0.76 67	0.13 30	0.12 8	0.22 33	0.50 25	1.04 29	0.72 27
Occlusion-TV-L1 [63]	35.3	0.09 26	0.26 35	0.07 14	0.22 30	0.74 37	0.18 34	0.51 51	1.15 55	0.21 40	0.18 37	0.91 38	0.10 36	0.87 42	1.25 40	0.72 42	0.47 29	1.38 39	0.36 39	0.10 4	0.12 8	0.11 2	0.83 57	1.78 60	0.96 52

Fig. 8. AAE and EPE on Middlebury test dataset. The proposed method ("Classic+CPF") is highlighted.

Table 5
Comparison of the original optical flow algorithms HS[4], BA [6], DN [7]. CP [34]with their improved variations integrated with our CPF method "HS+CPF", "BA+CPF", "DN+CPF" and "CP+CPF" in terms of AAE/EPE on some sequences from Middlebury dataset [2].

Method	Urban2	Urban3	Grove2	Grove3	RubberWhale	Venus
HS	4.060/0.459	**7.520/0.856**	2.853/0.204	6.809/0.690	3.798/0.118	**5.533/0.337**
HS+CPF	2.816/0.269	**6.076/0.730**	1.685/0.124	5.657/0.536	2.640/0.085	**3.754/0.253**
BA	2.965/0.376	**4.728/0.605**	2.492/0.172	6.496/0.660	3.156/0.097	**4.752/0.293**
BA+CPF	2.011/0.227	**4.049/0.513**	1.573/0.110	5.410/0.530	2.245/0.070	**3.354/0.237**
DN	3.865/0.581	**11.274/1.191**	2.683/0.195	7.014/0.742	4.184/0.130	**7.379/0.427**
DN+CPF	3.066/0.398	**6.373/0.819**	1.819/0.131	5.572/0.541	3.450/0.106	**6.923/0.380**
CP	2.645/0.357	**6.155/0.669**	2.952/0.211	6.718/0.656	3.988.0.145	**4.701/0.330**
CP+CPF	2.015/0.226	**4.516/0.535**	1.663/0.121	5.253/0.510	3.443/0.131	**3.627/0.273**

Table 6
Comparison of the original optical flow algorithms HS [4], BA [6], DN [7]. CP [34]with their improved variations integrated with our CPF method "HS+CPF", "BA+CPF", "DN+CPF" and "CP+CPF" in terms of AAE/EPE on some sequences from MIT dataset [35] and UCL GT Optical Flow Dataset v1.2 [3].

Method	Fish	Cameramotion	YoesmiteSun	Crates1	blow1Txtr1	drop1Txtr1	drop9Txtr2
HS	**30.726/1.351**	10.848/0.868	5.347/0.259	8.921/4.588	1.691/0.089	3.163/0.140	**9.380/2.700**
HS+CPF	**22.649/ 0.917**	8.129/0.673	3.798/0.201	6.049/4.325	0.797/0.050	1.991/0.100	**4.695/2.426**
BA	**23.897/1.050**	7.960/0.695	3.594/0.184	10.432/5.066	1.169/0.048	1.900/0.071	**7.517/2.922**
BA+CPF	**12.263/0.537**	5.726/0.529	2.778/0.150	7.172/4.840	0.410/0.020	1.183/0.048	**3.506/2.861**
DN	**26.096/1.109**	8.978/0.812	3.328/0.170	6.993/2.864	1.760/0.073	3.141/0.122	**17.660/5.084**
DN+CPF	**18.825/0.734**	6.479//0.623	2.469/0.130	4.797/2.105	1.003/0.046	2.290/0.088	**13.105/3.719**
CP	**21.457/0.898**	9.123/0.801	4.983/0.197	5.989/3.907	1.191/0.066	1.664/0.068	**8.435/1.966**
CP+CPF	**16.163/0.617**	6.672/0.635	2.768/0.130	4.495/3.378	0.546/0.043	0.866/0.040	**6.477/1.711**

nearly twice improved (from 23.897 to 12.263 and from 7.517 to 3.506, respectively) after using of our CPF method.

5.3. Evaluation on real sequences

To further evaluate the performance of the proposed "Classic+CPF" algorithm, we experiment with some real sequences from different datasets [2,8,36]. To the best of our knowledge, the test sequences in this work are the most widely selected compared to experiments in other optical flow works. These selected real sequences are obtained with different sensors, different frame rates and different scenes. They contain different illumination conditions (e.g. shaded, indoor, outdoor, dimmed light and bright light), different motions (large displacement, multiple moving objects) and different depth layers, and each contains a different challenge. Fig. 9 shows the results of Walking (indoor) and Backyard (outdoor) sequences on the Middlebury dataset [2]. For the Walking sequence, the "Classic+CPF" algorithm accurately estimates the motion of the human, the motion contours of his hands and the moving parts of his legs are all well represented. Moreover, his moving shadow is also clearly recovered. In addition, for the background, the edges of different objects are well preserved. The shape of the chair can be clearly identified from the oval flow at the bottom right of the flow field. For the Backyard sequence, the motion of the Walking boy and the twirl girl are well estimated. They can be easily distinguished from the flow field

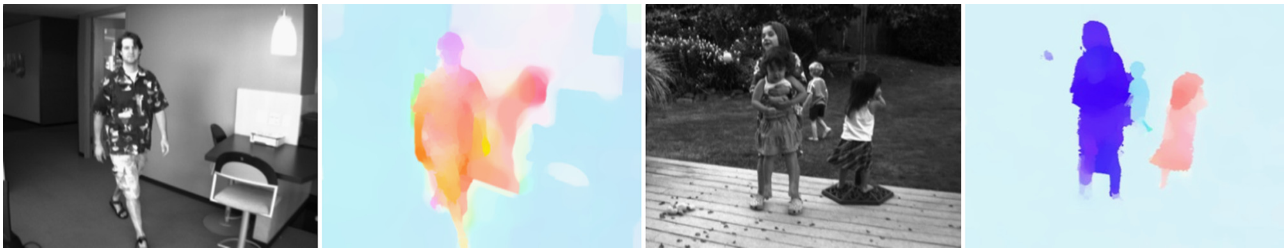


Fig. 9. Estimated optical flow fields on sequences Walking and Backyard with our "Classic+CPF" method. From left to right: frame 10 of Walking; the estimated flow field of Walking; frame 10 of Backyard; the estimated flow field of Backyard.

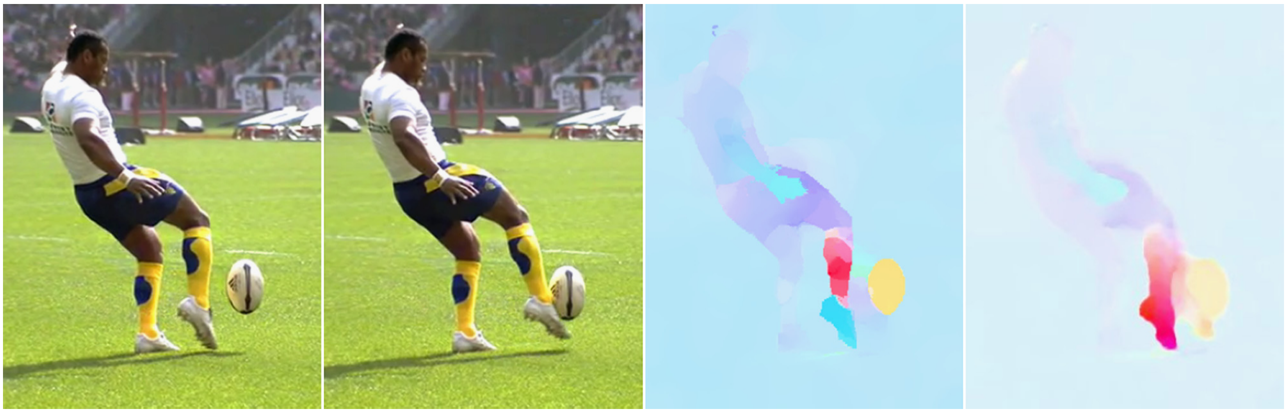


Fig. 10. Visual comparison of the optical flow results between our "Classic+CPF" method and the MDP-Flow2 [8] method on sequence Football (frames 3 and 4). From left to right: frame 3; frame 4; the estimated flow field of our method; the estimated flow field of the MDP-Flow2 method.

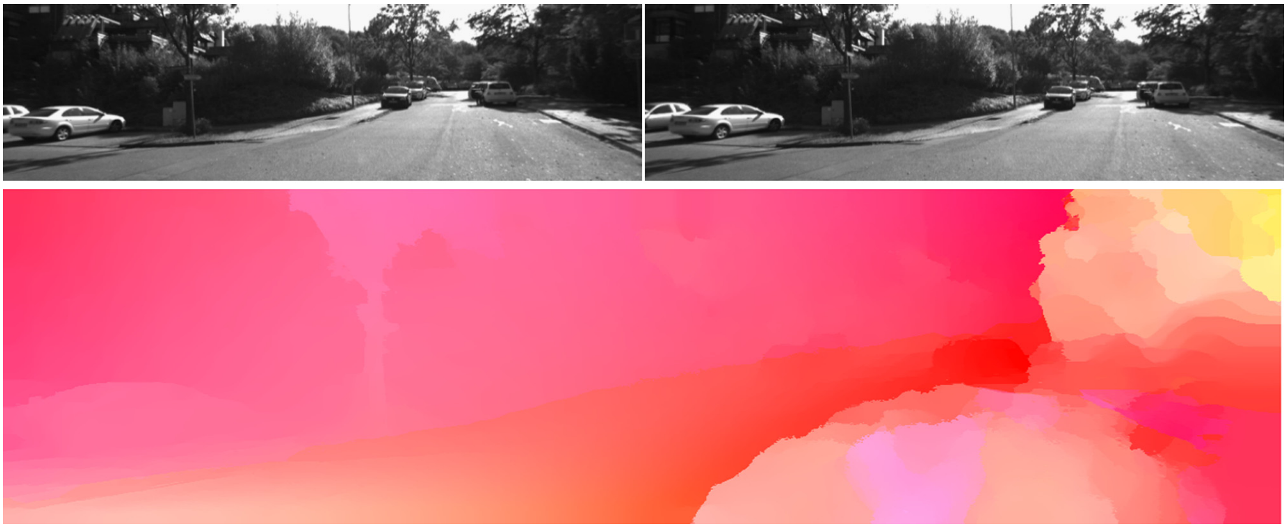


Fig. 11. The estimated optical flow field on sequence 000018 [36] with our "Classic+CPF" method. First line: from left to right: frame 10 (left) and frame 11 (right). Second line: the estimated flow field of our method.

by their sharp motion boundaries. For the two clasped girls, the motion boundaries of their head–neck region and the sleeve–arm region are also well estimated. Although there are some distinctive errors around their moving feet and legs, the shape and position can be easily recognized from the recovered flow field. Fig. 10 shows the results of the proposed "Classic+CPF" algorithm and the state-of-the-art MDP-Flow2 algorithm [8] (rank 3 at the Middlebury benchmark) of the Football sequence. It is difficult to estimate the motion of the player's right foot and the rapid moving small scale Football due to the large displacement and occlusion. The MDP-Flow2 algorithm can partially recover the motion of the right foot and the Football, but the recovered flow field is not clear, and also some errors are generated below the Football. In contrast, the quality of the computed flow field with the proposed "Classic+CPF" algorithm is much higher. The motion contours of the right hand, right foot and the Football are clearly reflected. Even the motion of the tiny left fingers which are located at the top of the head is recovered. Still, it disappears in the flow field of MDP-Flow2. Figs. 11–13, show three other examples on the KITTI dataset sequences 000018, 000024 and 000163 [36]. These sequences contain a type of complexity that is commonly encountered in outdoor environments. All the visual flow fields of these examples demonstrate that our method can remove outliers of the estimated flow field, while preserving the motion boundaries of the moving objects. Additionally, occlusions can be partially recovered. For instance, in Figs. 11 and 12, the complicated motions of the obscure trees are clearly reflected. From Fig. 13, we can see that even the moving shadow of the small scale rider and the bike is correctly recovered. Also the estimated motion contours of the small scale windows are sharply enough to be easily identified.

6. Conclusions

This paper presents a novel CPF method to improve the accuracy of variational optical flow algorithms. It contains two major steps: flow field edges and occlusions detection, and smoothing the classified flow field regions with different suitable filters. In the first step, flow field edges are accurately extracted with a nonlinear 3D spatial-scale Harris edge detector. The nonlinear 3D Harris edge detector is constructed by introducing scale information and replacing the Gaussian smoothing with a GBF.

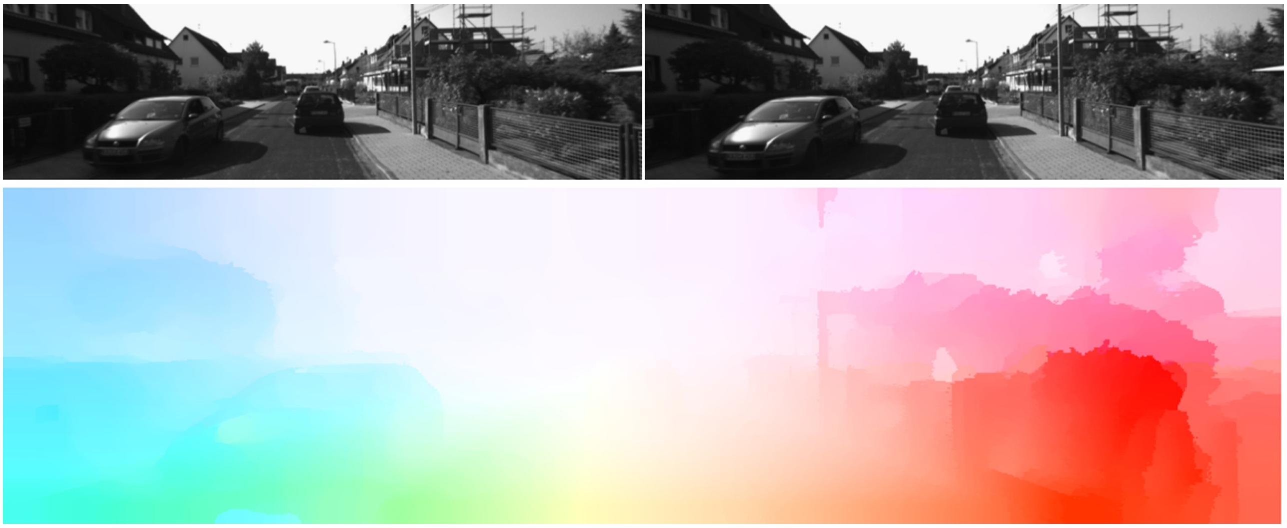


Fig. 12. The estimated optical flow field on sequence 000024 [36] with our "Classic+CPF" method. First line: from left to right: frame 10 (left) and frame 11 (right). Second line: the estimated flow field of our method.

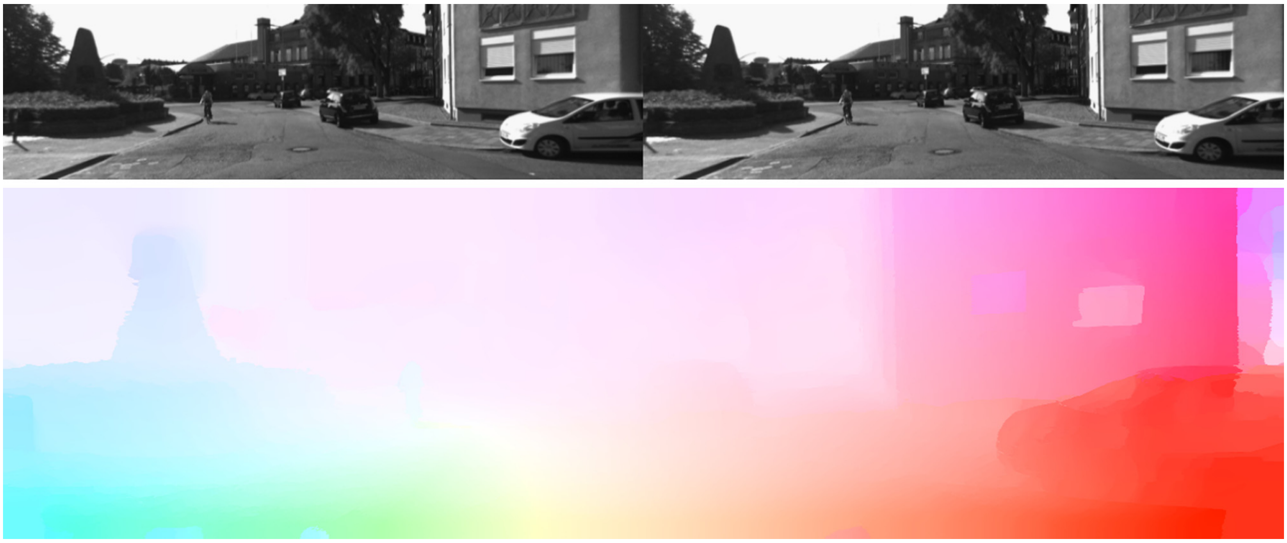


Fig. 13. The estimated optical flow field on sequence 000163 [36] with our "Classic+CPF" method. First line: from left to right: frame 10 (left) and frame 11 (right). Second line: the estimated flow field of our method.

Accurately edges identified, as their optimal scales can be determined due to the multi-scale technique, and the discontinuity blurring is reduced because the constructed nonlinear ST is adaptive to local structures. To improve the efficiency of the GBF based nonlinear ST, a new HGBGF smoothing approach is proposed by means of a new segmentation method based on spatial-scale gradient SNR. This segmentation method is used for classifying the ST elements into discontinuity regions and non-discontinuity regions. The time consuming GBF is only employed at discontinuity regions, while non-discontinuity regions still apply the Gaussian filter for smoothing. Furthermore, a piecewise occlusion detection approach is used for occlusion extraction. Second, the detected edges and occlusions, and the other flat regions of the flow field are post-filtered with the WMF, BF and the fast MF, respectively. Outliers are properly removed and discontinuities are well preserved. More important, occlusions are partially handled. In future work, we will focus on improving the CPF method from three promising aspects: (1) finding a way to adaptively set the filtering window size, since the fixed size based smoothing fails when meets with large displacements and large area occluded conditions; (2) proposing a scheme to adaptively change the ratio of different filters in the combination, since some sequences are more suitable to the BF while some sequences are more suitable to the WMF; and (3) reducing over-smoothing at the small structure regions by integrating more advanced elements into the filters like [33], since the WMF and the BF are not good enough to preserve small edges of the flow field.

Conflict of interest statement

None declared.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.patcog.2013.11.026.

References

[1] H.C. Longuet-Higgins, K. Prazdny, The interpretation of a moving retinal image, Proc. Roy. Soc. Lond. B 208 (1980) 385–397.

[2] S. Baker, D. Scharstein, J.P. Lewis, S. Roth, M.J. Black, R. Szeliski, A database and evaluation methodology for optical flow, Int. J. Comput. Vis. 92 (1) (2011) 1–31. (available at)⟨http://vision.middlebury.edu/flow/data/⟩.

[3] O. Aodha, A. Humayun, M. Pollefeys, G. Brostow, Learning a confidence measure for optical flow, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2013) 1107–1120. (available at)⟨http://visual.cs.ucl.ac.uk/pubs/flowConfidence/supp/index.html⟩.

[4] B. Horn, B. Schunck, Determining optical flow, Artif. Intell. 16 (1981) 185–203. (available at)⟨http://www.cs.brown.edu/∼dqsun/code/cvpr10_flow_code.zip⟩.

[5] N. Papenberg, A. Bruhn, T. Brox, S. Didas, J. Weickert, Highly accurate optic flow computation with theoretically justified warping, Int. J. Comput. Vis. 67 (2) (2006) 141–158.

[6] M.J. Black, P. Anandan, The robust estimation of multiple motions: Parametric and piecewise–smooth flow fields, Comput. Vis. Image Understanding 63 (1) (1996) 75–104. (available at).

[7] M. Drulea, S. Nedevschi, Total variation regularization of local-global optical flow, in: 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2011, pp. 318–323. Code resource available at ⟨http://www.cv.utcluj.ro/optical-flow.html?file=tl_files/cv/research/optical_flow/CLG-TV-matlab.zip⟩.

[8] L. Xu, J. Jia, Y. Matsushita, Motion detail preserving optical flow estimation, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1744–1757. (available at) ⟨http://www.cse.cuhk.edu.hk/leojia/projects/flow/⟩.

[9] H.H. Nagel, Constraints for the estimation of displacement vector fields from image sequences, in: International Joint Conference on Artificial Intelligence, 1983, pp. 945–951.

[10] H. Zimmer, A. Bruhn, J. Weickert, Optic flow in harmony, Int. J. Comput. Vis. 93 (3) (2011) 368–388.

[11] Z. Tu, W. Xie, W. Hürst, S. Xiong, Q. Qin, Weighted root mean square approach to select the optimal smoothness parameter of the variational optical flow algorithms, Opt. Eng. 51 (03) (2012) 037202–037209.

[12] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D: Nonlinear Phenom. 60 (1992) 259–268.

[13] C. Liu, W.T. Freeman, A high-quality video denoising algorithm based on reliable motion estimation, in: European Conference on Computer Vision, 2010, pp. 706–719.

[14] D. Sun, S. Roth, M.J. Black, Secrets of optical flow estimation and their principles, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2432–2439.

[15] C. Rabe, T. Müller, A. Wedel, U. Franke, Dense, robust and accurate motion field estimation from stereo image sequences in real-time, in: European Conference on Computer Vision, 2010, pp. 582–595.

[16] A. Wedel, T. Pock, C. Zach, D. Cremers, H. Bischof, An improved algorithm for TV-L1 optical flow, Statist. Geometrical Appl. Visual Motion Anal. 5064 (2008) 23–45.

[17] J. Xiao, H. Cheng, H. Sawhney, C. Rao, M. Isnardi, Bilateral filtering-based optical flow estimation with occlusion detection, in: European Conference on Computer Vision, 2006, pp. 211–224.

[18] P. Sand, S. Teller, Particle video: long-range motion estimation using point trajectories, Int. J. Comput. Vis. 80 (1) (2008) 72–91.

[19] Y Li, S Osher, A new median formula with applications to PDE based denoising, Commun. Math. Sci. 7 (3) (2009) 741–753.

[20] W. Förstner, E. Gülch, A fast operator for detection and precise location of distinct points, corners and circular features, in: Proceedings of the ISPRS Intercommission Workshop, 1987, pp. 281–305.

[21] T. Brox, J. Weickert, B. Burgeth, P. Mrázek, Nonlinear structure tensors, Image Vis. Comput. 24 (1) (2006) 41–55.

[22] L. Zhang, L. Zhang, D. Zhang, A multi-scale bilateral structure tensor based corner detector, in: Asian Conference on Computer Vision, 2009, pp. 618–627.

[23] D. Comaniciu, P. Meer, Mean shift: a robust approach towards feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.

[24] S. Heinrich, W.E. Snyder, Improved edge awareness in discontinuity preserving smoothing, CoRR 1103 (2011) 5808.

[25] F. Porikli, Constant time O(1) bilateral filtering, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[26] P.A.G. van Dorst, B.J. Janssen, L.M.J. Florack, B.M. ter Haar Romeny, Optic flow based on multi-scale anchor point movement and discontinuity-preserving regularization, Pattern Recognition 44 (9) (2011) 2057–2062.

[27] X. Ren, Multi-scale improves boundary detection in natural images, in: European Conference on Computer Vision, 2008, pp. 533–545.

[28] X. Liu, C. Wang, H. Yao, L. Zhang, The scale of edges, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp.462–469.

[29] C. Harris, M. Stephens, A combined corner and edge detector, in: 4th Alvey Vision Conference, 1988, pp. 147–151.

[30] J. J. Koenderink, The structure of images, Biol. Cybern., 1984.

[31] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2/3) (2005) 107–123.

[32] D. Martin, C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, color, and texture cues, IEEE Trans. Pattern Anal. Mach. Intell. 26 (5) (2004) 530–549.

[33] H.A. Rashwan, D. Puig, M.A. Garcia, Improving the robustness of variational optical flow through tensor voting, Comput. Vis. Image Understanding 116 (9) (2012) 953–966.

[34] A. Chambolle, T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging, J. Math. Imaging Vis. 40 (1) (2011) 120–145. (available at)⟨http://gpu4vision.icg.tugraz.at/index.php?content=downloads.php&id=46&field =Bin2⟩.

[35] Sequences available at ⟨http://people.csail.mit.edu/celiu/motionAnnotation/⟩.

[36] Sequences available at ⟨http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=flow⟩.

Zhigang Tu started his M.Phil., Ph.D. in image processing in the School of Electronic Information, Wuhan University, China, 2008. Since September 2011, he has been with the Multimedia and Geometry Group at Utrecht University, Netherlands. His current research interests include optical flow estimation, super-resolution construction, multimedia systems and technologies, human-computer interaction.

Nico van der Aa received his Ph.D. degree in numerical mathematics from Eindhoven University of Technology in 2007. In 2009 he joined Noldus Information Technology BV as a computer vision researcher in a scientific cooperation project with Utrecht University to develop algorithms for human behavior analysis. His expertise includes computer vision techniques for automatic human motion capturing, including people detection and articulated people tracking.

Coert Van Gemeren received his master's degree in cognitive artificial intelligence from the Humanities faculty of Utrecht University in 2012. After his graduate studies he became a Ph.D. candidate at the Science Faculty of Utrecht University. He is a computer vision researcher for the Interaction Technology group there, with a focus on the development of algorithms for interaction and mood classification in videos of groups of people.

Remco Veltkamp is full professor of Multimedia at Utrecht University, The Netherlands. His research interests are the analysis, recognition and retrieval of, and interaction with, music, images, and 3D objects and scenes, in particular the algorithmic and experimentation aspects. He has written over 150 refereed papers in reviewed journals and conferences, and supervised 15 Ph.D. theses. He was director of the national project GATE – Game Research for Training and Entertainment.