# Nonlocal Sparse and Low-rank Regularization for Optical Flow Estimation

Weisheng Dong, Guangming Shi, *Senior member, IEEE*, Xiaocheng Hu, and Yi Ma, *Fellow, IEEE*

***Abstract*—Designing an appropriate regularizer is of great importance for accurate optical flow estimation. Recent works exploiting the nonlocal similarity and the sparsity of the motion field have led to promising flow estimation results. In this paper we propose to unify these two powerful priors. To this end, we propose an effective flow regularization technique based on joint low-rank and sparse matrix recovery. By grouping similar flow patches into clusters, we effectively regularize the motion field by decomposing each set of similar flow patches into a low-rank component and a sparse component. For better enforcing the low-rank property, instead of using the convex nuclear norm, we use the $\log\det(\cdot)$ function as the surrogate of rank, which can also be efficiently minimized by iterative singular value thresholding. Experimental results on the Middlebury benchmark show that the performance of the proposed nonlocal sparse and low-rank regularization method is higher than (or comparable to) those of previous approaches that harness these same priors; and is competitive to current state-of-the-art methods.**

***Index Terms*—Optical flow, low-rank, sparse representation, nonlocal self-similarity.**

## I. Introduction

Optical flow estimation concerning the dense pixel correspondences between consecutive image frames is a fundamental problem in computer vision, and has many important applications in computer vision, such as 3D reconstruction, visual tracking, video enhancement, etc. Similar to other image restoration problems, optical flow estimation is also an ill-posed problem due to the aperture problem. Moreover, the problem is further complicated by many other issues, including: 1) the pixel intensity constancy assumption does not always hold; 2) optical flows in textureless regions are undefined; 3) pixels near motion boundaries may be occluded in the next frame, hence no valid matching exists. To successfully recover the dense optical flow, designing an appropriate flow regularization that incorporates the prior knowledge of the flow field is of great importance.

In the past decades, various optical flow estimation techniques have been proposed. Modern flow methods originating from the seminal work of Horn and Schunck (HS) [1] estimate the optical flow by minimizing an energy functional, which consists of a data term and a regularization term. The data term measures the matching error between the corresponding pixel pairs and the regularization term typically penalizes the horizontal and vertical deviations of the flow. In the work of [1], the $\ell_2$-norm was used for both data and regularization terms. Since the $\ell_2$-norm is not robust to data outliers caused by the brightness inconsistency and occlusions, it tends to generate over-smoothed flow results. In recent years, the flow estimation accuracy has been significantly improved by employing robust penalty functions for data and regularization terms. Various robust penalty functions, such as the convex $\ell_1$-norm (or its variants, e.g., Charbonnier function) and non-convex $\ell_p$-norms have been widely used [2]–[5]. Instead of using an arbitrarily selected robust function, attempts have also been made to learn the data term and the regularization term [6], [7]. In [6] the prior probability model of optical flow was learned using the Field-of-Expert model, and the learned model was used to regularize the spatial smoothness of the flow field. Furthermore, in [7] Sun et al. proposed to learn both the prior models of the data matching error and the flow field using the Gaussian Scale Mixture (GSM) model.

Motivated by the recent advances in nonlocal image restoration [8], nonlocal first order spatial regularization terms have also been proposed to exploit the nonlocal self-similarity [5], [9], [10]. In these works the nonlocal weights defining the similarity between the flow vector pairs were computed based on the color similarity. By exploiting the nonlocal redundancy, promising optical flow estimation results on the Middlebury dataset have been achieved. However, despite their effectiveness, only the pair-wise nonlocal dependencies are exploited in these works, leaving the rooms for further improvements. As another promising direction for flow field regularization, sparse representation based flow regularization techniques have also been proposed [11]–[13]. Compared to the conventional first-order spatial regularization, the sparse representation based regularizer can better model the local motion structures. It has been shown in [12] that the dictionary learned from a training flow dataset can lead to better flow estimation results than the off-the-shelf dictionaries. However, these sparsity-based flow estimation methods code each flow



The work was supported in part by the Major State Basic Research Development Program of China (973 Program) under Grant 2013CB329402, in part by the Natural Science Foundation of China under Grant 61227004, Grant 61471281, and Grant 61100154, in part by the Program for New Scientific and Technological Star of Shaanxi Province under Grant 2014KJXX-46, in part by the Fundamental Research Funds of the Central Universities of China under Grant BDY081424, and in part by the open project of Beijing Multimedia and Intelligent Software Key laboratory in Beijing University of Technology.

W. Dong and G. Shi are with School of Electronic Engineering, Xidian University, Xi'an, 710071, China (e-mail: wsdong@mail.xidian.edu.cn; gmshi@xidian.edu.cn).

X. Hu is with School of Information Science and Technology, University of Science and Technology of China, China (e-mail: hxc@mail.ustc.edu.cn).

Y. Ma is with the School of Information Science and Technology, ShanghaiTech University, China. He is also affiliated with the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, USA (e-mail: mayi@shanghaitech.edu.cn).

patch individually and thus cannot exploit the dependencies between nonlocal similar flow patches. It has also been known that the sparse decomposition under a redundant dictionary is potentially not robust.

*Contribution of this paper.* In this paper, to benefit both from the nonlocal spatial regularization and the sparsity-based regularization, we propose to unify the effective nonlocal and sparsity regularization into a unified variational framework. To exploit the nonlocal sparsity nature of optical flow, we propose to regularize the flow field by nonlocal flow patch grouping and low-rank and sparse matrices decomposition. Specifically, for each exemplar flow patch we group a set of similar flow patches to form a matrix, denoted by $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m]$, where $\boldsymbol{u}_i$ denote a flow patch vector. Since each matched flow patch should represent similar motion pattern, the flow patches without outliers should live in a low-dimension space and thus $\boldsymbol{U}$ should be well approximated by a low-rank matrix. Moreover, to deal with the outliers in the patches caused by occlusions or patch matching error, we propose to decompose $\boldsymbol{U}$ into a low-rank component and a sparse component. Such low-rank and sparse decomposition problem can be efficiently solved via convex optimization [14]–[16]. To better minimize the rank of the flow matrix, instead of using the nuclear norm, we propose to use the $\log\det(\cdot)$ function as a smooth surrogate of the rank and achieve better flow estimation performance. To the best of our knowledge, we are the first one to introduce the low-rank and sparse model for optical flow estimation.

Patch-based nonlocal sparse model has been successfully used for image denoising, e.g., the BM3D [17], the LSSC [18] and our previous CSR model [19], where image is denoised by patch matching and collaborative sparse coding. The nonlocal structure tensor TV regularization, which extends the local structure tensor TV (ST-TV) regularization [20], has also been proposed in [21] to promote a low-rank representation of the multi-component similar patches, showing very promising color image restoration results. However, the proposed nonlocal low-rank and sparse regularization model differs from previous nonlocal sparse models in that: 1) instead of using the simultaneous sparse model / ST-TV, the $log\det(\cdot)$ based low-rank model is used to effectively exploit the nonlocal sparsity of the motion field; 2) the nonlocal low-rank and sparsity model is integrated into the variational optical flow estimation framework, whereas the nonlocal sparsity-based image denoising as a post-processing of the estimated flow field cannot achieve satisfied flow estimation results; 3) coupled with the non-convex penalty functions, the resulted optimization problem is much more challenging.

Similar to the work in [11], [12] we also incorporate the first-order spatial regularization into the proposed flow estimation framework to further stabilize the estimation process. To deal with large motion, we embed the proposed algorithm into the conventional coarse-to-fine framework. Experimental results on Middlebury benchmark show that the performance of our method is higher than (or comparable) to those of the recently proposed nonlocal and sparsity-based flow estimation approaches [5], [10]–[13], [22], and is competitive with current state-of-the-art flow estimation methods.

## II. Related Work

In this section, we briefly review previous optical flow estimation works that are related to our work.

### A. Data term

Under the brightness constancy assumption, modern optical flow estimation methods estimate the dense optical flow by minimizing an energy functional, which is defined as

$$E(\boldsymbol{u}) = E_D(\boldsymbol{u}) + \lambda E_S(\boldsymbol{u}), \tag{1}$$

where $\boldsymbol{u} = [\boldsymbol{u}_x, \boldsymbol{u}_y]^\top \in \mathbb{R}^{2N}$ denotes the motion field to be estimated, $N$ denotes the number of image pixels and $\lambda$ is a regularization parameter. To simplify the notations, we will use $\boldsymbol{u}_z$ to refer to $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$, where $z \in \{x, y\}$ denotes both the horizontal and vertical directions. The data term that measures the matching error between the consecutive frame images $I_1$ and $I_2$ is defined as $E_D(\boldsymbol{u}) = \sum_{\boldsymbol{x}} \phi_D(|I_1(\boldsymbol{x}) - I_2(\boldsymbol{x}+\boldsymbol{u})|^2)$, where $\phi_D$ is a penalty function, and $\boldsymbol{x} \in \mathbb{Z}^2$ denotes a 2D coordinate. Various penalty functions have been proposed, including the convex $\ell_2$ norm, the $\ell_1$ norm or its variant (e.g., $\sqrt{x^2+\epsilon^2}$), and the non-convex $\ell_p$ norm or its variant, called Generalized Charbonnier (GC) function $(x^2 + \epsilon^2)^\alpha$ [5]. It has been shown in [5] that the GC function with $\alpha = 0.45$ generally leads to the best results.

When the change of the motion flow $\boldsymbol{u}$ is small, we can approximate the prediction residual $\rho(\boldsymbol{x}) = I_2(\boldsymbol{x}+\boldsymbol{u}) - I_1(\boldsymbol{x})$ by linearizing about the current estimate of $\boldsymbol{u}$ as $\rho(\boldsymbol{x}) = I_t + \nabla I_2^\top(\boldsymbol{u} - \boldsymbol{u}^0)$, where $I_t = I_2(\boldsymbol{x} + \boldsymbol{u}^0) - I_1(\boldsymbol{x})$ denotes the temporal derivation, $\nabla I_2$ denotes the derivation of $I_2$ at $\boldsymbol{x}+\boldsymbol{u}^0$, and $\boldsymbol{u}^0$ denotes the current estimate of $\boldsymbol{u}$.

In addition to the brightness constancy, other constancy constraints can also been imposed, e.g., the gradient constancy constraint [23]. The gradient constancy constraint has advantages in overcoming the illumination change but does a poor job in smooth regions. Recent studies [24], [25] show that the combination of the brightness constancy and the gradient constancy constraints can lead to better flow estimation results.

### B. Spatial regularization term

Due to the well-known aperture problem, optical flow estimation is an ill-posed inverse problem and prior knowledge of the flow field is required to regularize the flow field. The widely used first-order spatial regularization techniques that penalize the deviation of the flow field in both horizontal and vertical directions can be defined as $E_S(\boldsymbol{u}) = \sum_{\boldsymbol{x}} \phi_S(|\nabla \boldsymbol{u}_x(\boldsymbol{x})|^2 + |\nabla \boldsymbol{u}_y(\boldsymbol{x})|^2)$, where $\phi_S$ is a penalty function. The quadratic first-order spatial regularization tends to generate over-smoothed flow field. Combined with the robust penalty functions, the motion discontinuities can be better preserved. To better deal with the data outliers, in [13] Chen *et al.* proposed to decompose the flow gradients into the sparse component and the non-sparse component, which represent the motion discontinuities and dense Gaussian noise, respectively. By sparse and non-sparse modeling, their approach can better preserve the motion boundaries.

Attempts have been made to construct nonlocal flow regularization term. In [5], [9], [10], [22] the nonlocal TV model

was used to regularize the flow field. Let the spatial locations of the nonlocal similar neighbor of pixel at $\boldsymbol{x}$ be defined as $\boldsymbol{x} \sim \boldsymbol{q}, \boldsymbol{q} \in \mathbb{Z}^2$. Then, the nonlocal TV regularizer can be expressed as

$$E_{Nonl}(\boldsymbol{u}) = \sum_{\boldsymbol{x}\sim\boldsymbol{q}} w_{\boldsymbol{q}} \phi_N(\boldsymbol{u}_{\boldsymbol{x}} - \boldsymbol{u}_{\boldsymbol{q}}), \tag{2}$$

where $w_{\boldsymbol{q}}$ are the bilateral filter coefficients computed using the color pixels, and $\phi_N$ denotes a robust penalty function. By exploiting the nonlocal redundancy, promising flow estimation results on the Middlebury benchmark dataset have been achieved. However, only the first order nonlocal spatial statistics are exploited in the above nonlocal TV model.

Motivated by the compressive sensing theory [26], sparsity-based regularization models have also been proposed for flow estimation [11], [12]. These approaches regularize the flow field by pursuing a sparse representation of the flow using either an off-the-shelf dictionary [11] or a learned dictionary [12]. In the patch-based sparsity regularization, the flow field is first partitioned into multiple flow patches of size $\sqrt{n} \times \sqrt{n}$ with overlapping region. Then each flow patch is sparsely coded with respect to an overcomplete dictionary. The sparsity-based regularization term can be formulated as

$$E_{Spar}(\boldsymbol{u}) = \mu \sum_{i=1}^{P} \|\boldsymbol{R}_i \boldsymbol{u} - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \tau \sum_i \|\boldsymbol{\alpha}_i\|_1, \tag{3}$$

where $\boldsymbol{R}_i \in \mathbb{R}^{2n\times 2N}$ denotes a matrix extracting a patch flow at position $\boldsymbol{x}_i \in \mathbb{Z}^2$, $\boldsymbol{D} = [\boldsymbol{D}_x \ \boldsymbol{0}; \boldsymbol{0} \ \boldsymbol{D}_y] \in \mathbb{R}^{2n\times 2p}$ $(p \geq n)$ and $\boldsymbol{\alpha}_i \in \mathbb{R}^{2p}$ denote the dictionary and the representation coefficient vector of the $i$-$th$ flow patch, respectively. In general, the flow patches can be extracted uniformly along the horizontal and vertical directions with overlapping region. With an appropriately designed dictionary, the sparsity-based approaches have shown improvements in preserving the local motion structures. However, current sparsity-based flow estimation methods code each flow patch individually and cannot utilize structural correlation between the sparse codes.

To effectively exploit the local and nonlocal structural dependencies of the flow, in this paper we propose a novel optical flow regularization model by nonlocal sparse and low-rank matrix decomposition. The key idea is to group a set of similar flow patches for each exemplar flow patch and decompose the matrix formed by the similar patches into a low-rank component and a sparse error component, as explained in the following sections.

## III. Low-rank and Sparse Regularization of the Optical Flow Field

There are two key components in the proposed low-rank and sparse regularization technique: (1) Flow patch grouping; (2) decomposing the grouped flow matrices into low-rank and sparse components.

### A. Flow patch grouping

As a first step of our method, we need to group a set of similar flow patches for each extracted exemplar flow patch. Since the ground truth flow field is unavailable, patch matching using the truth flow field is impossible. Instead, we can perform flow patch grouping using initial recovered optical flow. However, due to estimation error such patch grouping may not be robust. In this paper, we propose to perform the flow patch matching by using the color images. The basic assumption is that similar local image structures tend to have similar motions. Though this assumption may not be always true, our experimental results show that patch matching using color images works well in our experiments.

Let $\boldsymbol{u}_i \in \mathbb{R}^{2n}$ denote local flow patches of size $\sqrt{n} \times \sqrt{n}$ (horizontal component) centered at positions $\boldsymbol{x}_i$, $i = 1, 2, \cdots, P$. Let $\boldsymbol{p}_i \in \mathbb{R}^{3n}$ denote local color image patches centered at positions $\boldsymbol{x}_i$. For each extracted color image patch $\boldsymbol{p}_i$, we search for a set of similar patches across the whole image by patch matching. To save computational complexity, we can limit the search region to a large neighborhood (e.g., a $40 \times 40$ window). The patch matching can be expressed as

$$G_i = \{j | \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2^2 \leq T\}, \tag{4}$$

where $T$ is a pre-defined threshold and $G_i$ contains the coordinates of the similar patches. Equivalently, we can also select the similar patch if it is within the first $m$ ($m = 30$ in our implementation) most similar patches to $\boldsymbol{p}_i$. After patch matching, we can then obtain a set of flow patches similar to $\boldsymbol{u}_i$, denoted by $\boldsymbol{u}_j, j \in G_i$.

### B. Low-rank and sparse matrices decomposition

Using the patch matching approach described above, for each exemplar flow patch $\boldsymbol{u}_i$ we can collect a set of similar flow patches to form a matrix $\boldsymbol{U}_i = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m] \in \mathbb{R}^{2n\times m}$, where $\boldsymbol{u}_j = [\boldsymbol{u}_{x,j}, \boldsymbol{u}_{y,j}]^\top \in \mathbb{R}^{2n}$, $j = 1, 2, \cdots, m$. To simplify the low-rank and sparse matrix decomposition problem, we treat the horizontal and vertical component separately. Let $\boldsymbol{U}_{z,i} \in \mathbb{R}^{n\times m}$, $z \in \{x, y\}$ denote both the horizontal and vertical components of $\boldsymbol{U}_i$. Then we will use $\boldsymbol{U}_{z,i}$ as an example to present the low-rank and sparse matrix decomposition method. We assume that these flow patches represent similar motion structures and thus they should live in a low dimensional subspace. In practice, the matrix $\boldsymbol{U}_{z,i}$ may be corrupted by some errors. If the matrix $\boldsymbol{U}_{z,i}$ only contains Gaussian noise, then $\boldsymbol{U}_{z,i}$ can be accurately approximated by a low-rank matrix using principle component analysis (PCA). However, due to the illuminance change and occlusions, the estimated flow may also contain large sparse errors. Since PCA is sensitive to large errors, directly approximate $\boldsymbol{U}_{z,i}$ with PCA may not be robust. Motivated by the recent works on low-rank matrix recovery [14]–[16], we propose to model the flow matrix $\boldsymbol{U}_{z,i}$ as: $\boldsymbol{U}_{z,i} = \boldsymbol{L}_{z,i} + \boldsymbol{S}_{z,i} + \boldsymbol{W}_{z,i}$, where $\boldsymbol{L}_{z,i}$ denotes the low-rank matrix, $\boldsymbol{S}_{z,i}$ denotes the sparse error matrix and $\boldsymbol{W}_{z,i}$ denotes the Gaussian noise matrix. Note that $\boldsymbol{S}_{z,i}$ may also contain the misalignment error of the flow patch vectors. $\boldsymbol{L}_{z,i}$ and $\boldsymbol{S}_{z,i}$ can be recovered by solving the following minimization problem:

$$\min_{\boldsymbol{L}_{z,i}, \boldsymbol{S}_{z,i}} \text{rank}(\boldsymbol{L}_{z,i}) + \lambda \|\boldsymbol{S}_{z,i}\|_1, \ \text{ s.t. } \|\boldsymbol{U}_{z,i} - \boldsymbol{L}_{z,i} - \boldsymbol{S}_{z,i}\|_F^2 \leq \sigma_n^2, \tag{5}$$

where $\|\cdot\|_F^2$ denotes the Frobenious norm and $\sigma_n^2$ is the variance of the Gaussian noise. Since the rank-minimization problem is in general an NP-hard problem, we cannot directly solve Eq.(5) efficiently. As a convex surrogate of the rank, the nuclear norm $\|\cdot\|_*$ (sum of the singular values) has been widely used to approximate the rank minimization problem. Using the nuclear norm, the rank minimization problem can be easily solved by the singular value thresholding (SVT) [27], with good theoretical guarantee of correctness [15]. Nevertheless, recent studies show that empirically non-convex surrogates of the rank minimization may lead to better recovery results. In this paper, instead of using the nuclear norm, we consider a smooth but non-convex surrogate of the rank. In [28] it has been shown that for a symmetric positive semidefinite matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$, the rank minimization problem can be approximated by minimizing the following functional:

$$E(\boldsymbol{X},\varepsilon) \doteq \log\det(\boldsymbol{X}+\varepsilon\boldsymbol{I}), \tag{6}$$

where $\varepsilon$ is a small regularization parameter. Note that this function $E(\boldsymbol{X},\varepsilon)$ approximates the sum of the logarithm of the singular values (up to a scale). Thus, the function $E(\boldsymbol{X},\varepsilon)$ is not convex, though it is smooth.

To minimize the rank of $\boldsymbol{L}_{z,i}$, we slightly modify Eq.(6) as

$$L(\boldsymbol{L}_{z,i},\varepsilon) \doteq \log\ \det((\boldsymbol{L}_{z,i}^\top\boldsymbol{L}_{z,i})^{1/2}+\varepsilon\boldsymbol{I}), \tag{7}$$

which is a surrogate function of rank$(\boldsymbol{L}_{z,i})$, as obtained by setting $\boldsymbol{X} = (\boldsymbol{L}_{z,i}^\top\boldsymbol{L}_{z,i})^{1/2}$. Now we can solve for $\boldsymbol{L}_{z,i}$ by minimizing

$$\min_{\boldsymbol{L}_{z,i},\boldsymbol{S}_{z,i}} L(\boldsymbol{L}_{z,i},\varepsilon)+\lambda\|\boldsymbol{S}_{z,i}\|_1,\ \text{s.t.}\ \|\boldsymbol{U}_{z,i}-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2 \le \sigma_n^2. \tag{8}$$

Instead of solving this constrained minimization problem, we solve it in its Lagragian form, as

$$\min_{\boldsymbol{L}_{z,i},\boldsymbol{S}_{z,i}} \frac{1}{2\mu}\|\boldsymbol{U}_{z,i}-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2+L(\boldsymbol{L}_{z,i},\varepsilon)+\lambda\|\boldsymbol{S}_{z,i}\|_1. \tag{9}$$

With a proper selection of $\mu$, Eq.(9) is equivalent to Eq.(8). For each exemplar flow patch, we can obtain the matrix $\boldsymbol{U}_{z,i}$ and decompose it into two components by solving Eq.(9).

### C. Proposed flow regularization

Now the problem is how to estimate the whole flow field using this patch-wise low-rank and sparse regularization. In this paper we propose the following objective function for optical flow estimation

$$\begin{aligned} &E(\boldsymbol{u},\boldsymbol{L}_{x,i},\boldsymbol{S}_{x,i},\boldsymbol{L}_{y,i},\boldsymbol{S}_{y,i}) = E_D(\boldsymbol{u})+ \\ &\sum_{z\in\{x,y\}}\sum_{i=1}^{P}\big\{\frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2+ \\ &L(\boldsymbol{L}_{z,i},\varepsilon)+\lambda\|\boldsymbol{S}_{z,i}\|_1\big\}, \end{aligned} \tag{10}$$

where $\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z = [\boldsymbol{R}_1\boldsymbol{u}_z,\boldsymbol{R}_2\boldsymbol{u}_z,\cdots,\boldsymbol{R}_m\boldsymbol{u}_z]$ denotes the matrix formed by the set of similar patches grouped using the method described above.

The data term $E_D(\boldsymbol{u})$ is defined as

$$E_D(\boldsymbol{u}) = \sum_{\boldsymbol{x}}\phi_D(|I_t+I_x d\boldsymbol{u}_x+I_y d\boldsymbol{u}_y|^2), \tag{11}$$

where $I_x$ and $I_y$ denotes the horizontal and vertical gradients of image $I$, respectively, and $d\boldsymbol{u}_x = \boldsymbol{u}_x-\boldsymbol{u}_x^0$ and $d\boldsymbol{u}_y = \boldsymbol{u}_y-\boldsymbol{u}_y^0$ denotes the small change of $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$, respectively. For the robust penalty function $\phi_D$, we use the generalized Charbonnier penalty function (with $\alpha = 0.45$) [5].

The proposed low-rank and sparse regularization in Eq.(10) exploits both the nonlocal redundancies and the joint sparsity of these similar local flow patches. However, the assumption that similar image structures undergo similar motion may become invalid. Incorrect patch clustering will increase the rank of the matrix $\boldsymbol{U}_i$ and thus decrease the performance of the proposed method. For robustness, we further incorporate the first-order spatial regularization into our objective function, leading to the following objective function

$$\begin{aligned} &E(d\boldsymbol{u},\boldsymbol{L}_{x,i},\boldsymbol{S}_{x,i},\boldsymbol{L}_{y,i},\boldsymbol{S}_{y,i}) = \sum_{\boldsymbol{x}}\phi_D(|I_t+I_x d\boldsymbol{u}_x+I_y d\boldsymbol{u}_y|^2) \\ &+\sum_{z\in\{x,y\}}\sum_{i=1}^{P}\big\{\frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2+L(\boldsymbol{L}_{z,i},\varepsilon) \\ &+\lambda\|\boldsymbol{S}_{z,i}\|_1\big\}+\sum_{\boldsymbol{x}}\eta\phi_S(|\nabla\boldsymbol{u}_x(\boldsymbol{x})|^2+|\nabla\boldsymbol{u}_y(\boldsymbol{x})|^2), \end{aligned} \tag{12}$$

where $\eta$ is a constant controlling the contribution of the first-order spatial regularization and the robust penalty function $\phi_S$ is also the generalized Charbonnier penalty function.

## IV. Optimization Algorithm

A standard approach to minimize the objective function of Eq.(12) is to alternatively optimizing the flow filed $\boldsymbol{u}$ and the set of low-rank matrices $\boldsymbol{L}_{z,i}$ and the sparse matrices $\boldsymbol{S}_{z,i}$.

### A. Low-rank and sparse matrix optimization

For an initial estimate of the flow $\boldsymbol{u}$, we first cluster the flow patches into many groups using the method described in section III.A, and then solve for the low-rank matrix $\boldsymbol{L}_{z,i}$ and the sparse matrix $\boldsymbol{S}_{z,i}$ by minimizing:

$$\min_{\boldsymbol{L}_{z,i},\boldsymbol{S}_{z,i}} \frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2+L(\boldsymbol{L}_{z,i},\varepsilon)+\lambda\|\boldsymbol{S}_{z,i}\|_1, \tag{13}$$

which can be solved by alternatively optimizing $\boldsymbol{L}_{z,i}$ and $\boldsymbol{S}_{z,i}$. For fixed $\boldsymbol{L}_{z,i}$, $\boldsymbol{S}_{z,i}$ can be solved by minimizing

$$\min_{\boldsymbol{S}_{z,i}} \frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2+\lambda\|\boldsymbol{S}_{z,i}\|_1, \tag{14}$$

which can be easily solved by soft-thresholding:

$$\hat{\boldsymbol{S}}_{z,i} = \mathcal{S}_{\lambda\mu}(\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z-\boldsymbol{L}_{z,i}), \tag{15}$$

where $\mathcal{S}_{\lambda\mu}$ denotes the soft-thresholding operator with threshold $\lambda\mu$.

For fixed $\boldsymbol{S}_{z,i}$, $\boldsymbol{L}_{z,i}$ is solved by minimizing

$$\min_{\boldsymbol{L}_{z,i}} \frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i\boldsymbol{u}_z-\boldsymbol{L}_{z,i}-\boldsymbol{S}_{z,i}\|_F^2+L(\boldsymbol{L}_{z,i},\varepsilon). \tag{16}$$

Note that $L(\boldsymbol{L}_i, \varepsilon)$ is approximately the sum of the logarithm of the singular values (up to a scale). Therefore, Eq.(16) can be rewritten as

$$\min_{\boldsymbol{L}_{z,i}} \frac{1}{2\mu}\|\boldsymbol{Y}_{z,i} - \boldsymbol{L}_{z,i}\|_F^2 + \sum_{j=1}^{n_0} \log(\sigma_j(\boldsymbol{L}_{z,i}) + \varepsilon). \quad (17)$$

where $\boldsymbol{Y}_{z,i} = \tilde{\boldsymbol{R}}_i \boldsymbol{u}_z - \boldsymbol{S}_{z,i}$, $n_0 = min\{n, m\}$ and $\sigma_j(\boldsymbol{L}_{z,i})$ denotes the $j^{th}$ singular value of $\boldsymbol{L}_{z,i}$. For simplicity, we use $\sigma_j$ to denote the $j^{th}$ singular value of $\boldsymbol{L}_{z,i}$. Though $\sum_{j=1}^{n} \log(\sigma_j + \varepsilon)$ is non-convex, we can efficiently solve it using a local minimization method [28] for a local minimum. Let $f(\boldsymbol{\sigma}) = \sum_{j=1}^{n} \log(\sigma_j + \varepsilon)$. We can approximate $f(\boldsymbol{\sigma})$ by first order Taylor expansion, as

$$f(\boldsymbol{\sigma}) = f(\boldsymbol{\sigma}^{(k)}) + \langle \nabla f(\boldsymbol{\sigma}^{(k)}), \boldsymbol{\sigma} - \boldsymbol{\sigma}^{(k)} \rangle, \quad (18)$$

where $\boldsymbol{\sigma}^{(k)}$ is the solution obtained in the $k^{th}$ iteration. Thus, one can solve Eq.(17) by iteratively minimizing

$$\boldsymbol{L}_{z,i}^{(k+1)} = \underset{\boldsymbol{L}_{z,i}}{\operatorname{argmin}} \frac{1}{2\mu}\|\boldsymbol{Y}_{z,i} - \boldsymbol{L}_{z,i}\|_F^2 + \sum_{j=1}^{n_0} \frac{\sigma_j}{\sigma_j^{(k)} + \varepsilon}, \quad (19)$$

where we have used $\nabla f(\boldsymbol{\sigma}^{(k)}) = \sum_{j=1}^{n_0} \frac{1}{\sigma_j^{(k)} + \varepsilon}$ and ignored the constants in Eq.(18). Eq.19 can be solved by the singular value thresholding algorithm:

$$\boldsymbol{L}_{z,i}^{(k+1)} = \boldsymbol{D}\mathcal{S}_{\boldsymbol{\tau}^{(k)}}(\tilde{\boldsymbol{\Sigma}})\boldsymbol{V}^\top \quad (20)$$

where $\boldsymbol{D}\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}^\top$ is the SVD of $\boldsymbol{Y}_{z,i}$, $\mathcal{S}_{\boldsymbol{\tau}^{(k)}}$ denotes the soft threshold operator with threshold $\tau_j^{(k)} = \mu/(\sigma_j^{(k)} + \varepsilon)$. In our implementation, we choose $\sigma_j^{(0)} = 1$ and the first iteration is equivalent to solving a un-weighted nuclear norm minimization problem.

A special case of the $\log\det(\cdot)$ leads to the popular reweighted $\ell_1$-norm when the matrix $\boldsymbol{L}_{z,i}$ is a vector [29]. It has been shown in [29] that the reweighted $\ell_1$-norm is better than $\ell_1$-norm in approximating the $\ell_0$-norm. Our experimental results show that the $\log\det(\cdot)$ can lead to better flow estimation results than the nuclear norm.

### B. Optical flow optimization

For fixed $\boldsymbol{L}_i$ and $\boldsymbol{S}_i$, we estimate the flow field by minimizing

$$\begin{aligned}(d\boldsymbol{u}_x, d\boldsymbol{u}_y) &= \underset{d\boldsymbol{u}_x, d\boldsymbol{u}_y}{\operatorname{argmin}} \sum_{\boldsymbol{x}} \phi_D(|I_t + I_x d\boldsymbol{u}_x + I_y d\boldsymbol{u}_y|^2) \\ &+ \sum_{z\in\{x,y\}} \sum_{i=1}^{P} \frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i(d\boldsymbol{u}_z + \boldsymbol{u}_z^0) - \boldsymbol{L}_{z,i} - \boldsymbol{S}_{z,i}\|_F^2 \\ &+ \sum_{\boldsymbol{x}} \eta\phi_S(|\nabla(\boldsymbol{u}_x + \boldsymbol{u}_x^0)|^2 + |\nabla(\boldsymbol{u}_y + \boldsymbol{u}_y^0)|^2),\end{aligned} \quad (21)$$

where we have replaced $\boldsymbol{u}_z$ with $d\boldsymbol{u}_z + \boldsymbol{u}_z^0$. The minimization of Eq. (21) is generally difficult due to the non-convexity of the robust functions $\phi_D$ and $\phi_S$ used in Eq.(21). For robustness, we adopt the graduate non-convexity (GNC) scheme [7] to find a local minimum. Specifically, we start the iterative process by replacing the non-convex robust functions $\phi_D$ and $\phi_S$ with quadratic functions. The initial solution of the quadratic objective function serves as the starting point for the next non-convex minimization stage.

In practice, we first obtain an initial estimate of the flow field by using the first-order TV regularizer. Using the initial estimate of the flow, we perform the flow patch grouping for each exemplar and then solve for the sparse and low-rank matrices by minimizing Eqs. (14) and (16). The obtained $\boldsymbol{L}_{z,i}$ and $\boldsymbol{S}_{z,i}$ are then used to improve the flow estimation by solving Eq.(21). With an improved flow estimation, the low-rank and sparse matrices can be updated. Such process is iterated until convergence. We start this iterative process with a large value of $\mu$, and then gradually decrease the value of $\mu$ by setting $\mu^{(k+1)} = \gamma\mu^{(k)}$, where $\gamma < 1$ is a pre-determined constant. For fixed $\boldsymbol{L}_{z,i}$ and $\boldsymbol{S}_{z,i}$, we propose to use the iterative reweighted least-square (IRLS) algorithm to solve Eq.(21). The details of the IRLS for solving Eq.(21) are presented in Appendix A. The overall algorithm for flow estimation via sparse and low-rank regularization (FESL) are summarized in Algorithm 1.

We empirically found that the algorithm converges even when the inner loop only executes one iteration. This will much save the computational complexity of the proposed algorithm. Thus, we set $J = 1$ in our implementation.

## V. Experimental Evaluation

In this section we evaluate the proposed flow estimation method on the widely used Middlebury benchmark dataset [30]. To deal with the illuminance change issue, we use the structure decomposition method in [31] to pre-process the input image frames. To handle large motions we integrate the proposed flow estimation method into a coarse-to-fine warping framework. This also helps to effectively optimizing the large scale non-convex objective function [32]. We use the downsampling factor of $0.8$ to construct the pyramids. For each pyramid we conduct $4$ warping steps. In each warping step, we execute $K = 30$ iterations in the outer loop of Algorithm 1. In our implementation, we tuned the algorithm parameters on the Middlebury training set for better performance and use the same parameters for all experiments presented in this paper. The parameters are set as follows: $\mu = 1$, $\gamma = 0.83$, $\eta = 0.5$ and $\lambda = 0.45$. The flow patch size $\sqrt{n} \times \sqrt{n}$ is set to $5 \times 5$, and total $m = 30$ similar flow patches are collected for each exemplar flow patch. To speed up the computation, we only use exemplars every four pixels along both horizontal and vertical directions. This also much saves the computational complexity of the proposed algorithm without decrease of the performance.

### A. Effectiveness of the low-rank and sparse regularization

To verify the effectiveness of the proposed sparse and low-rank regularization, we implement four variants of the proposed flow estimation methods. Let "SR" denote the flow estimation methods using spatial regularization, which minimizes the following objective function

$$\begin{aligned}E(d\boldsymbol{u}) &= \sum_{\boldsymbol{x}} \phi_D(|I_t + I_x d\boldsymbol{u}_x + I_y d\boldsymbol{u}_y|^2) \\ &+ \sum_{\boldsymbol{x}} \eta\phi_S(|\nabla \boldsymbol{u}_x(\boldsymbol{x})|^2 + |\nabla \boldsymbol{u}_y(\boldsymbol{x})|^2).\end{aligned} \quad (22)$$

**Algorithm 1** Flow estimation via sparse and low-rank regularization

- Initialization:
  - Compute an initial flow field using the first-order spatial regularization;
  - Clustering color image patches into many groups for $G_i$;
- Outer loop: for $k = 1, 2, ....., K$
  - Inner loop (Low-rank and Sparse decomposition): for $j = 1, 2, \cdots, J$
    - Update the sparse matrix $\boldsymbol{S}_{z,i}$ via Eq.(15);
    - Update the low-rank matrix $\boldsymbol{L}_{z,i}$ via Eq.(20);
  - Update the optical flow $\boldsymbol{u}$ by solving Eq.(21);
  - $\mu \rightarrow \gamma\mu$.

Let "SR-LR-NN" denote the flow estimation method that regularizes the flow field by minimizing the following objective function

$$E(d\boldsymbol{u}, \boldsymbol{L}_{x,i}, \boldsymbol{L}_{y,i}) = \sum_{\boldsymbol{x}} \phi_D(|I_t + I_x d\boldsymbol{u}_x + I_y d\boldsymbol{u}_y|^2) + \sum_{z\in\{x,y\}} \sum_{i=1}^{P} \{\frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i \boldsymbol{u}_z - \boldsymbol{L}_{z,i}\|_F^2 + ||\boldsymbol{L}_{z,i}||_* + \sum_{\boldsymbol{x}} \eta\phi_S(|\nabla \boldsymbol{u}_x(\boldsymbol{x})|^2 + |\nabla \boldsymbol{u}_y(\boldsymbol{x})|^2)\}. \quad (23)$$

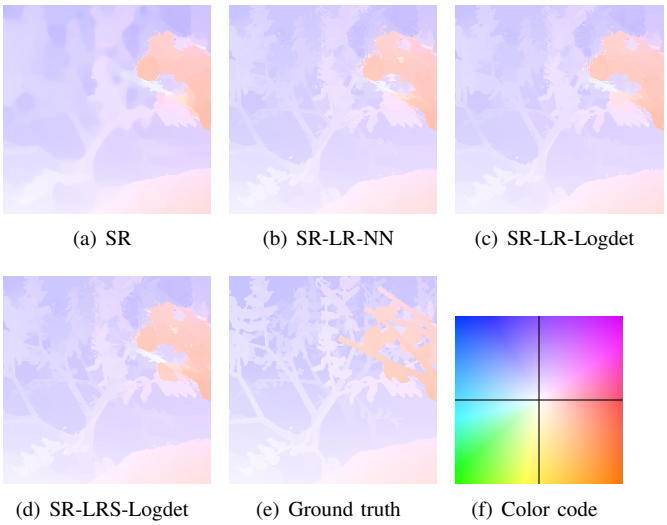
where $||\cdot||_*$ denotes the nuclear norm. The third variant, denoted by "SR-LR-Logdet", regularizes the flow field by solving the following objective function

$$E(d\boldsymbol{u}, \boldsymbol{L}_{x,i}, \boldsymbol{L}_{y,i}) = \sum_{\boldsymbol{x}} \phi_D(|I_t + I_x d\boldsymbol{u}_x + I_y d\boldsymbol{u}_y|^2) + \sum_{z\in\{x,y\}} \sum_{i=1}^{P} \{\frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i \boldsymbol{u}_z - \boldsymbol{L}_{z,i}\|_F^2 + L(\boldsymbol{L}_{z,i}, \varepsilon) + \sum_{\boldsymbol{x}} \eta\phi_S(|\nabla \boldsymbol{u}_x(\boldsymbol{x})|^2 + |\nabla \boldsymbol{u}_y(\boldsymbol{x})|^2)\}, \quad (24)$$

The last variant denoted by "SR-LRS-Logdet", solves for the flow field by minimizing the following energy

$$E(d\boldsymbol{u}, \boldsymbol{L}_{x,i}, \boldsymbol{S}_{x,i}, \boldsymbol{L}_{y,i}, \boldsymbol{S}_{y,i}) = \sum_{\boldsymbol{x}} \phi_D(|I_t + I_x d\boldsymbol{u}_x + I_y d\boldsymbol{u}_y|^2) + \sum_{z\in\{x,y\}} \sum_{i=1}^{P} \{\frac{1}{2\mu}\|\tilde{\boldsymbol{R}}_i \boldsymbol{u}_z - \boldsymbol{L}_{z,i} - \boldsymbol{S}_{z,i}\|_F^2 + L(\boldsymbol{L}_{z,i}, \varepsilon) + \lambda\|\boldsymbol{S}_{z,i}\|_1 + \sum_{\boldsymbol{x}} \eta\phi_S(|\nabla \boldsymbol{u}_x(\boldsymbol{x})|^2 + |\nabla \boldsymbol{u}_y(\boldsymbol{x})|^2)\}. \quad (25)$$

The "SR-LRS-Logdet" method can be implemented by **Algorithm 1**, while the other three variants can be implemented by modifying slightly **Algorithm 1**. Applying the four variants to the training part of Middlebury dataset, we show the average endpoint error (AEPE) and average angle error (AAE) results in Table I. From Table I, we can see that the low-rank regularization leads to better results than the TV-based spatial regularization. By replacing the nuclear norm with the logdet function as a surrogate of the rank, further improvements can be obtained. By combining the sparse term, the SR-LRS-logdet methods can achieve further improvements. However, the improvements are not significant. This is mainly because the data term in our approach still cannot handle well the occlusions, though the sparse term can well handle the outliers in the flow field. Some parts of the color coded flow fields of training sequence *Grove3* are shown in Fig. 1. To visualize the optical flow, the color-coding scheme of [30] are used. The color and intensity code the orientations and magnitudes of optical flow, respectively.



(a) SR (b) SR-LR-NN (c) SR-LR-Logdet

(d) SR-LRS-Logdet (e) Ground truth (f) Color code

Fig. 1. Parts of color coded flow results of sequences *Grove3* using the four variants of the proposed method. Optical flow fields are visualized using the color code in (f).

### *B. Comparison with other state-of-the-arts*

We first conducted the experiments using the training part of Middlebury dataset, and compared to other flow estimation methods, including the improved TV-$l_1$ method [31], the nonlocal TV method (denoted as NL-TV-NCC) [10] and the classic+NL method [5] that exploits nonlocal redundancy, the sparse and nonsparse method [13], the learned sparse method (LSM) [12] that exploits the local sparsity with learned redundant dictionaries, and the efficient nonlocal regularization method (denoted as Efficient-NL) [22] [1]. By developing an efficient optimization algorithm, the Efficient-NL method [22] can efficiently minimize the nonlocal regularization terms of any given spatial extent, achieving much better results than previous nonlocal regularization methods.

[1]We thank the authors of [5], [22] for providing their source code or experimental results.

TABLE I
AEPE AND AAE RESULTS OF THE FIVE VARIANTS OF THE PROPOSED METHOD ON THE MIDDLEBURY TRAINING DATASET.

| method | AEPE results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RubberWhale | Hydrangea | Dimetrodon | Grove2 | Grove3 | Urban2 | Urban3 | Venus | average |
| SR | 0.081 | 0.159 | 0.121 | 0.108 | 0.534 | 0.246 | 0.410 | 0.228 | 0.236 |
| SR-LR-NN | 0.074 | 0.167 | 0.115 | 0.097 | 0.452 | 0.252 | 0.350 | 0.228 | 0.217 |
| SR-LR-Logdet | 0.070 | 0.161 | 0.118 | 0.095 | 0.434 | 0.253 | 0.345 | 0.223 | 0.212 |
| SR-LRS-Logdet | 0.067 | 0.155 | 0.122 | 0.092 | 0.423 | 0.246 | 0.344 | 0.223 | 0.209 |
| | AAE results | | | | | | | | |
| SR | 2.566 | 1.857 | 2.348 | 1.567 | 5.425 | 1.970 | 2.972 | 3.050 | 2.719 |
| SR-LR-NN | 2.299 | 2.029 | 2.228 | 1.398 | 4.607 | 1.945 | 2.559 | 3.028 | 2.512 |
| SR-LR-Logdet | 2.144 | 1.952 | 2.296 | 1.356 | 4.377 | 1.982 | 2.529 | 2.977 | 2.452 |
| SR-LRS-Logdet | 2.057 | 1.909 | 2.369 | 1.313 | 4.294 | 1.904 | 2.524 | 2.916 | 2.411 |

Table II shows the average endpoint error (AEPE) and average angle error (AAE) results obtained by the competing methods, respectively. From Table II, we can see that the proposed low-rank and sparse method achieves substantial improvements over the TV method, NL-TV-NCC method and the classic+NL method. The proposed method also outperforms the sparsity based methods, i.e., the Sparse-NonSparse method and the LSM method. The proposed method is comparable to the Efficient-NL method with respect to AEPE metric, and is better than the Efficient-NL method in terms of AAE metric. The proposed method outperforms Efficient-NL method on relative smooth motions (e.g., *Venus*, *RubberWhale*), while becomes worse than Efficient-NL method on larger motion (i.e., *Urban2*), in which case the proposed method may fail to find enough similar flow patches via image patch matching. In addition to the better AEPE and AAE performances, the proposed regularization technique also enjoys advantages in preserving the motion boundaries. Fig. 2 shows parts of the color coded flow fields of training sequences *Venus*, *Grove3* and *Urban3*. From Fig. 2, we can see that the Efficient-NL method preserves the details of the motion structures much better than the Classic+NL method. The proposed method performs slightly better than Efficient-NL method in preserving large motion boundaries (e.g., *Venus* and *Urban3*), while is slightly inferior to Efficient-NL method in recovering fine details of motion structures (e.g., *Grove3*). In Fig. 3 we give the color coded flow results of 8 sequences in the training Middlebury set.

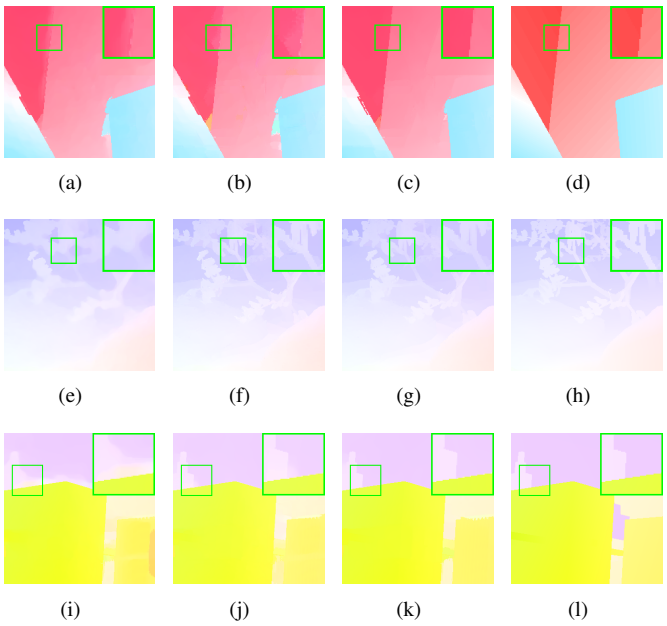



Fig. 2. Parts of color coded flow results of sequences *Venus* (top row), *Grove3* (middle row), and *Urban3* (bottom row). From left to right: Classic+NL [5]; Efficient-NL [22]; **Proposed**; Ground truth. **(Please view this figure on screen)**

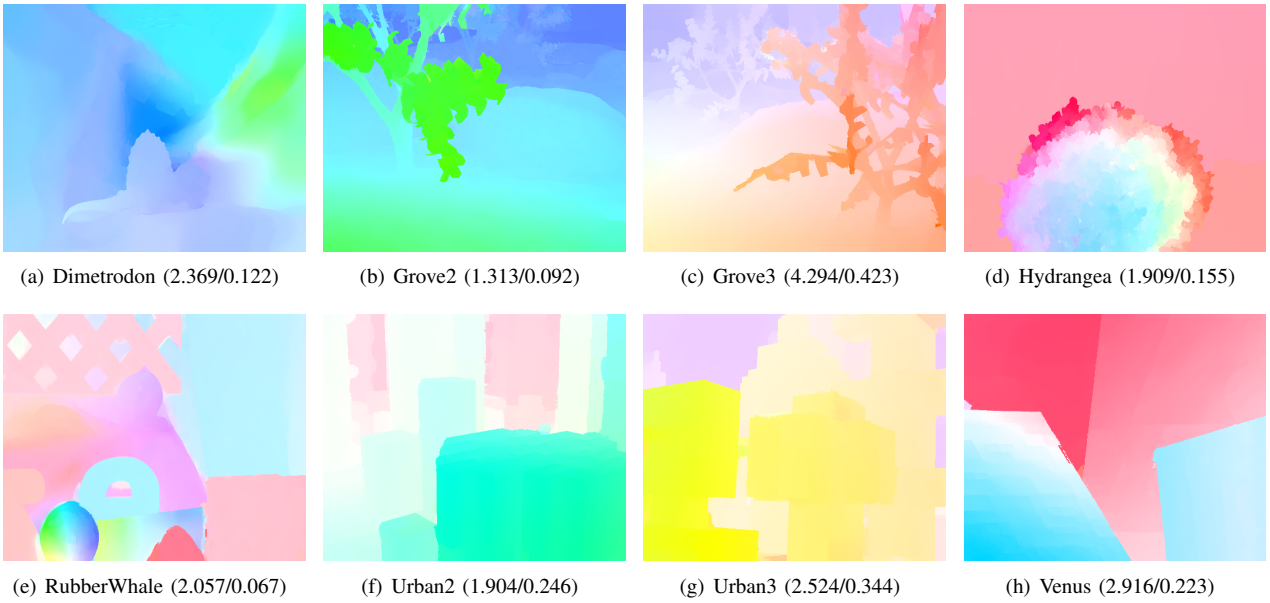
### C. Overall performance assessment

We also ran the proposed flow estimation method on the Middleblury test dataset and submitted our results to Middlebury website. Fig. 4 shows the comparison results with other leading flow estimation methods. At the moment of paper submission, our results rank nine according to AAE and seven according to AEPE. From Fig. 4, we can see that our approach outperforms the methods that exploit similar priors for flow regularization, including the nonlocal TV methods [5] and the learned sparse method [12]. Our approach performs better than the Efficient-NL method [22] with respect to AEPE metric, while worse than Efficient-NL method with respect to AAE metric. Note that the top methods nLayers [33] and its previous version Layers ++ [34] use a graphic model to explicitly address the depth order issue and thus can handle the occlusions much better; moreover nLayers [33] uses multiple frame images, which can provide further improvements; MDP-Flow [25] addresses the issue of large displacement motions by using extended flow initialization. In Fig. 5, we show the color coded flow results of 8 sequences in the Middlebury test dataset.

In this work, we intentionally stay within a very basic and simple formulation, and do not explicitly model or address additional factors such as depth-order, multi-layer multi-frame, and large displacement. Our goal is to develop a global flow regularizer by exploiting the nonlocal self-similarity and sparsity, and to demonstrate its effectiveness. Despite the rankings, using this regularization *alone*, our method already performs closely to the top systems. If needed, the proposed low-rank and sparse flow regularizer can be combined with

TABLE II
AEPE AND AAE RESULTS COMPARISON ON THE MIDDLEBURY TRAINING DATASET.

| method | AEPE results | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | RubberWhale | Hydrangea | Dimetrodon | Grove2 | Grove3 | Urban2 | Urban3 | Venus | average |
| TV-$l_1$ [31] | 0.092 | 0.147 | 0.190 | 0.154 | 0.665 | 0.319 | 0.630 | 0.260 | 0.335 |
| NL-TV-NCC [10] | 0.080 | 0.150 | 0.160 | 0.140 | 0.540 | 0.350 | 0.430 | 0.290 | 0.268 |
| Classic+NL [5] | 0.077 | 0.151 | 0.117 | 0.098 | 0.464 | **0.210** | 0.421 | 0.232 | 0.221 |
| LSM [12] | 0.072 | 0.151 | 0.129 | 0.105 | 0.473 | 0.221 | 0.375 | 0.235 | 0.220 |
| Sparse-NonSparse [13] | 0.070 | **0.140** | **0.120** | 0.100 | 0.480 | **0.210** | 0.370 | 0.230 | 0.215 |
| Efficient-NL [22] | 0.074 | 0.155 | 0.119 | 0.097 | 0.425 | 0.217 | 0.351 | 0.243 | 0.210 |
| **Proposed** | **0.067** | 0.155 | 0.122 | **0.092** | **0.423** | 0.246 | **0.344** | **0.223** | **0.209** |
| | AAE results | | | | | | | | |
| NL-TV-TNCC [10] | 2.550 | 1.920 | 3.210 | 1.940 | 6.050 | 2.920 | 3.360 | 4.270 | 3.278 |
| Classic+NL [5] | 2.402 | 1.824 | 2.280 | 1.410 | 4.930 | 2.029 | 3.164 | 3.289 | 2.666 |
| LSM [12] | 2.285 | **1.803** | 2.541 | 1.511 | 5.005 | 2.004 | 2.599 | 3.297 | 2.631 |
| Efficient-NL [22] | 2.345 | 1.924 | 2.330 | 1.374 | 4.408 | 2.021 | 2.602 | 3.487 | 2.561 |
| **Proposed** | **2.057** | 1.909 | **2.369** | **1.313** | **4.294** | **1.904** | **2.524** | **2.916** | **2.416** |



(a) Dimetrodon (2.369/0.122) (b) Grove2 (1.313/0.092) (c) Grove3 (4.294/0.423) (d) Hydrangea (1.909/0.155)

(e) RubberWhale (2.057/0.067) (f) Urban2 (1.904/0.246) (g) Urban3 (2.524/0.344) (h) Venus (2.916/0.223)

Fig. 3. Color coded flow results of Middlebury training set. The AAE and AEPE values are given in brackets (AAE/AEPE).

the other more detailed flow estimation techniques (e.g., those in [25], [33], [34]) to further improve the flow estimation accuracy on the test dataset.

### D. Parameters selection

To evaluate the sensitivity of the proposed algorithm with respect to the main parameters, i.e., $\mu$, $\lambda$ and $\eta$, we varied the parameters $\mu$, $\lambda$ and $\eta$. Fig. 6 plots the curves of the average AAE and EPE metrics as functions of the parameters on the Middlebury training dataset.

The parameter $\mu$ controls the low-rank and sparse regularization of the optical flow. An excessively small $\mu$ yields a small penalty of the low-rank and sparse regularization and thus fails to regularize the optical flow estimation process, while an excessively large $\mu$ leads to overly heavy penalty of the low-rank and sparse regularization and hence decreases the estimation accuracy. As shown in Fig. 6 (a) and (b), the performance of the proposed algorithm is insensitive to the choices of $\mu$ when $0.7 < \mu < 8.4$. The parameter $\lambda$ controls the contributions of the sparse regularization term. A small $\lambda$ yields more nonzero components in the sparse matrixes, while large $\lambda$ yields less nonzero components in the sparse matrixes. From Fig. 6 (c) and (d), we can see that the performance of the proposed method didn't vary significantly when varying $\lambda$ in the range of $(0.14, 0.74)$. The parameter $\eta$ controls the contribution of the first-order spatial regularization. From Fig. 6 (e) and (f), it can be seen that the integration of the spatial regularization with a small regularization parameter can slightly improve the performance. However, an excessive large $\eta$ decreases the performance.

### E. Running time

The proposed algorithms were implemented with Matlab language. The running time of the proposed algorithms on

| Average angle error | avg. rank | Army (Hidden texture) GT all | Army im0 all | Army im0 disc | Army im0 untext | Army im1 all | Army im1 disc | Army im1 untext | Mequon (Hidden texture) GT all | Mequon im0 all | Mequon im0 disc | Mequon im0 untext | Mequon im1 all | Mequon im1 disc | Mequon im1 untext | Schefflera (Hidden texture) GT all | Schefflera im0 all | Schefflera im0 disc | Schefflera im0 untext | Schefflera im1 all | Schefflera im1 disc | Schefflera im1 untext | Wooden (Hidden texture) GT all | Wooden im0 all | Wooden im0 disc | Wooden im0 untext | Wooden im1 all | Wooden im1 disc | Wooden im1 untext | Grove (Synthetic) GT all | Grove im0 all | Grove im0 disc | Grove im0 untext | Grove im1 all | Grove im1 disc | Grove im1 untext | Urban (Synthetic) GT all | Urban im0 all | Urban im0 disc | Urban im0 untext | Urban im1 all | Urban im1 disc | Urban im1 untext | Yosemite (Synthetic) GT all | Yosemite im0 all | Yosemite im0 disc | Yosemite im0 untext | Yosemite im1 all | Yosemite im1 disc | Yosemite im1 untext | Teddy (Stereo) GT all | Teddy im0 all | Teddy im0 disc | Teddy im0 untext | Teddy im1 all | Teddy im1 disc | Teddy im1 untext |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NN-field [73] | 4.8 | 2.89 3 | 8.13 7 | 2.11 1 | 2.10 3 | 7.15 4 | 1.77 6 | 2.27 1 | 5.59 1 | 1.61 3 | 1.58 1 | 8.52 1 | 0.79 1 | 2.35 2 | 3.05 3 | 1.60 1 | 1.89 1 | 5.20 1 | 1.37 1 | 2.43 26 | 3.70 25 | 1.95 20 | 1.01 1 | 2.25 1 | 0.53 1 |
| nLayers [57] | 7.8 | 2.80 1 | 7.42 1 | 2.20 3 | 2.71 14 | 7.24 5 | 2.55 31 | 2.61 4 | 6.24 4 | 2.45 28 | 2.30 3 | 12.7 5 | 1.16 3 | 2.30 1 | 3.02 1 | 1.70 2 | 2.62 5 | 6.95 2 | 2.09 3 | 2.29 19 | 3.46 13 | 1.89 17 | 1.38 6 | 3.06 9 | 1.29 6 |
| MDP-Flow2 [70] | 8.6 | 3.23 18 | 7.93 4 | 2.60 12 | 1.92 1 | 6.64 1 | 1.52 1 | 2.46 3 | 5.91 3 | 1.56 2 | 3.05 20 | 15.8 22 | 1.51 20 | 2.77 10 | 3.50 7 | 2.16 10 | 2.86 7 | 8.58 7 | 2.70 14 | 2.00 9 | 3.50 18 | 1.59 8 | 1.28 3 | 2.67 3 | 0.89 3 |
| ADF [67] | 10.9 | 2.98 6 | 8.32 10 | 2.28 4 | 2.27 5 | 8.35 10 | 1.81 7 | 3.55 16 | 9.74 18 | 2.17 14 | 3.15 26 | 16.8 28 | 1.29 6 | 2.64 7 | 3.55 8 | 1.81 3 | 3.02 8 | 9.08 8 | 2.38 6 | 2.29 19 | 3.48 15 | 2.07 23 | 1.34 4 | 3.03 7 | 1.11 4 |
| Layers++ [37] | 13.0 | 3.11 8 | 8.22 8 | 2.79 21 | 2.43 8 | 7.02 3 | 2.24 16 | 2.43 2 | 5.77 2 | 2.18 17 | 2.13 2 | 9.71 2 | 1.15 2 | 2.35 2 | 3.02 1 | 1.96 5 | 3.81 25 | 11.4 21 | 3.22 30 | 2.74 37 | 4.01 40 | 2.35 29 | 1.45 8 | 3.05 8 | 1.79 16 |
| Efficient-NL [60] | 13.6 | 2.99 7 | 8.23 9 | 2.28 4 | 2.72 15 | 8.95 14 | 2.25 18 | 3.81 20 | 9.87 20 | 2.07 12 | 2.77 14 | 14.3 13 | 1.46 14 | 2.61 5 | 3.48 6 | 1.96 5 | 3.31 15 | 8.33 5 | 2.59 10 | 2.60 31 | 3.75 26 | 2.54 34 | 1.60 12 | 3.02 6 | 1.66 11 |
| LME [72] | 13.8 | 3.15 12 | 8.04 6 | 2.31 7 | 1.95 2 | 6.65 2 | 1.59 3 | 4.03 25 | 9.31 16 | 4.57 51 | 2.69 10 | 13.6 7 | 1.42 9 | 2.85 16 | 3.61 11 | 2.42 21 | 3.47 21 | 12.8 26 | 3.17 27 | 2.12 13 | 3.53 21 | 1.73 11 | 1.34 4 | 2.75 4 | 1.18 5 |
| ALD-Flow [68] | 14.1 | 2.82 2 | 7.86 3 | 2.16 2 | 2.84 19 | 10.1 20 | 1.86 9 | 3.73 18 | 10.4 21 | 1.67 5 | 3.10 22 | 16.8 28 | 1.28 5 | 2.69 8 | 3.60 10 | 1.85 4 | 2.79 6 | 11.3 20 | 2.32 5 | 2.07 10 | 3.25 5 | 3.10 52 | 2.03 23 | 5.11 24 | 1.94 18 |
| FESL [75] | 14.1 | 2.96 5 | 7.70 2 | 2.54 10 | 3.26 35 | 10.4 22 | 2.56 32 | 3.25 9 | 8.39 9 | 2.17 14 | 2.56 5 | 13.2 6 | 1.31 7 | 2.57 4 | 3.40 5 | 2.12 9 | 2.60 4 | 7.65 3 | 2.30 4 | 2.64 35 | 4.22 45 | 2.47 31 | 1.75 16 | 3.49 14 | 1.71 13 |
| IROF++ [58] | 14.6 | 3.17 14 | 8.69 14 | 2.61 13 | 2.79 16 | 9.61 17 | 2.33 20 | 3.43 12 | 8.86 13 | 2.38 22 | 2.87 16 | 14.8 16 | 1.52 21 | 2.74 9 | 3.57 9 | 2.19 11 | 3.20 13 | 9.70 14 | 2.71 15 | 1.96 8 | 3.45 12 | 1.22 8 | 1.80 17 | 4.06 18 | 2.50 25 |
| SCR [74] | 14.8 | 3.12 9 | 8.48 11 | 2.59 11 | 2.95 25 | 10.4 22 | 2.35 21 | 3.19 7 | 8.09 7 | 2.43 26 | 2.63 6 | 13.9 10 | 1.35 8 | 2.81 12 | 3.64 12 | 2.30 13 | 3.02 8 | 8.29 4 | 2.39 8 | 2.77 40 | 3.79 29 | 2.89 45 | 1.39 7 | 2.85 5 | 1.60 10 |
| Sparse-NonSparse [56] | 16.0 | 3.14 11 | 8.75 16 | 2.76 20 | 3.02 28 | 10.6 28 | 2.43 25 | 3.45 14 | 8.96 14 | 2.36 20 | 2.66 8 | 13.7 9 | 1.42 9 | 2.85 16 | 3.75 17 | 2.33 15 | 3.28 14 | 9.40 11 | 2.73 16 | 2.42 25 | 3.31 6 | 2.69 36 | 1.47 9 | 3.07 10 | 1.66 11 |
| **Average endpoint error** | **avg. rank** | **Army (Hidden texture) GT all** | **Army im0 all** | **Army im0 disc** | **Army im0 untext** | **Army im1 all** | **Army im1 disc** | **Army im1 untext** | **Mequon (Hidden texture) GT all** | **Mequon im0 all** | **Mequon im0 disc** | **Mequon im0 untext** | **Mequon im1 all** | **Mequon im1 disc** | **Mequon im1 untext** | **Schefflera (Hidden texture) GT all** | **Schefflera im0 all** | **Schefflera im0 disc** | **Schefflera im0 untext** | **Schefflera im1 all** | **Schefflera im1 disc** | **Schefflera im1 untext** | **Wooden (Hidden texture) GT all** | **Wooden im0 all** | **Wooden im0 disc** | **Wooden im0 untext** | **Wooden im1 all** | **Wooden im1 disc** | **Wooden im1 untext** | **Grove (Synthetic) GT all** | **Grove im0 all** | **Grove im0 disc** | **Grove im0 untext** | **Grove im1 all** | **Grove im1 disc** | **Grove im1 untext** | **Urban (Synthetic) GT all** | **Urban im0 all** | **Urban im0 disc** | **Urban im0 untext** | **Urban im1 all** | **Urban im1 disc** | **Urban im1 untext** | **Yosemite (Synthetic) GT all** | **Yosemite im0 all** | **Yosemite im0 disc** | **Yosemite im0 untext** | **Yosemite im1 all** | **Yosemite im1 disc** | **Yosemite im1 untext** | **Teddy (Stereo) GT all** | **Teddy im0 all** | **Teddy im0 disc** | **Teddy im0 untext** | **Teddy im1 all** | **Teddy im1 disc** | **Teddy im1 untext** |
| MDP-Flow2 [70] | 4.9 | 0.08 4 | 0.21 2 | 0.07 10 | 0.15 1 | 0.48 1 | 0.11 1 | 0.20 2 | 0.40 2 | 0.14 1 | 0.15 10 | 0.80 17 | 0.08 6 | 0.63 8 | 0.93 9 | 0.43 8 | 0.26 1 | 0.76 2 | 0.23 3 | 0.11 9 | 0.12 6 | 0.17 10 | 0.38 2 | 0.79 1 | 0.44 2 |
| NN-field [73] | 5.8 | 0.08 4 | 0.22 7 | 0.05 1 | 0.17 4 | 0.55 4 | 0.13 5 | 0.19 1 | 0.39 1 | 0.15 3 | 0.09 1 | 0.48 1 | 0.05 1 | 0.41 1 | 0.61 1 | 0.20 1 | 0.52 31 | 0.64 1 | 0.26 5 | 0.13 23 | 0.13 20 | 0.20 19 | 0.35 1 | 0.83 2 | 0.21 1 |
| ADF [67] | 10.4 | 0.08 4 | 0.22 7 | 0.06 2 | 0.18 5 | 0.62 9 | 0.14 9 | 0.29 15 | 0.71 19 | 0.17 8 | 0.16 19 | 0.91 28 | 0.07 2 | 0.69 16 | 1.03 16 | 0.47 11 | 0.43 12 | 0.91 5 | 0.28 7 | 0.12 15 | 0.12 6 | 0.20 19 | 0.43 3 | 0.88 4 | 0.63 9 |
| LME [72] | 10.7 | 0.08 4 | 0.22 7 | 0.06 2 | 0.15 1 | 0.49 2 | 0.11 1 | 0.30 18 | 0.64 12 | 0.31 51 | 0.15 10 | 0.78 15 | 0.09 15 | 0.66 11 | 0.96 11 | 0.53 18 | 0.33 3 | 1.18 18 | 0.28 7 | 0.12 15 | 0.12 6 | 0.18 13 | 0.44 4 | 0.91 6 | 0.61 7 |
| IROF++ [58] | 10.9 | 0.08 4 | 0.23 11 | 0.07 10 | 0.21 17 | 0.68 17 | 0.17 18 | 0.28 12 | 0.63 11 | 0.19 18 | 0.15 10 | 0.73 11 | 0.09 15 | 0.60 5 | 0.89 5 | 0.42 7 | 0.43 12 | 1.08 13 | 0.31 15 | 0.10 4 | 0.12 6 | 0.12 4 | 0.47 9 | 0.98 12 | 0.68 16 |
| Layers++ [37] | 11.0 | 0.08 4 | 0.21 2 | 0.07 10 | 0.19 9 | 0.56 5 | 0.17 18 | 0.20 2 | 0.40 2 | 0.18 11 | 0.13 2 | 0.58 3 | 0.07 2 | 0.48 2 | 0.70 2 | 0.33 3 | 0.47 21 | 1.01 8 | 0.33 22 | 0.15 40 | 0.14 33 | 0.24 34 | 0.46 7 | 0.88 4 | 0.72 19 |
| nLayers [57] | 11.4 | 0.07 1 | 0.19 1 | 0.06 2 | 0.22 21 | 0.59 6 | 0.19 30 | 0.25 7 | 0.54 6 | 0.20 27 | 0.15 10 | 0.84 20 | 0.08 6 | 0.53 3 | 0.78 3 | 0.34 4 | 0.44 16 | 0.84 3 | 0.30 13 | 0.13 23 | 0.13 20 | 0.20 19 | 0.47 9 | 0.97 11 | 0.67 13 |
| ALD-Flow [68] | 15.3 | 0.07 1 | 0.21 2 | 0.06 2 | 0.19 9 | 0.64 14 | 0.13 5 | 0.30 18 | 0.73 20 | 0.15 3 | 0.17 24 | 0.92 32 | 0.07 2 | 0.78 23 | 1.14 23 | 0.59 22 | 0.33 3 | 1.30 25 | 0.21 1 | 0.12 15 | 0.12 6 | 0.28 48 | 0.54 23 | 1.19 25 | 0.73 22 |
| FESL [75] | 15.4 | 0.08 4 | 0.21 2 | 0.07 10 | 0.25 37 | 0.75 29 | 0.19 30 | 0.27 8 | 0.61 9 | 0.18 11 | 0.14 4 | 0.68 4 | 0.08 6 | 0.61 7 | 0.89 5 | 0.44 9 | 0.47 21 | 1.03 10 | 0.32 18 | 0.14 31 | 0.15 42 | 0.25 37 | 0.50 17 | 0.96 9 | 0.63 9 |
| SCR [74] | 15.5 | 0.08 4 | 0.23 11 | 0.07 10 | 0.22 21 | 0.71 21 | 0.17 18 | 0.27 8 | 0.60 8 | 0.19 18 | 0.14 4 | 0.73 11 | 0.08 6 | 0.63 8 | 0.92 8 | 0.44 9 | 0.51 29 | 1.08 13 | 0.33 22 | 0.15 40 | 0.13 20 | 0.29 53 | 0.47 9 | 0.93 7 | 0.67 13 |
| COFM [59] | 15.5 | 0.08 4 | 0.26 26 | 0.06 2 | 0.18 5 | 0.62 9 | 0.14 9 | 0.30 18 | 0.74 22 | 0.19 18 | 0.15 10 | 0.86 22 | 0.07 2 | 0.79 24 | 1.14 23 | 0.74 38 | 0.35 6 | 0.87 4 | 0.28 7 | 0.14 31 | 0.12 6 | 0.28 48 | 0.49 13 | 0.94 8 | 0.71 18 |
| TC-Flow [46] | 15.7 | 0.07 1 | 0.21 2 | 0.06 2 | 0.15 1 | 0.59 6 | 0.11 1 | 0.31 22 | 0.78 25 | 0.14 1 | 0.16 19 | 0.86 22 | 0.08 6 | 0.75 21 | 1.11 22 | 0.54 19 | 0.42 11 | 1.40 32 | 0.25 4 | 0.11 9 | 0.12 6 | 0.29 53 | 0.62 27 | 1.35 27 | 0.93 38 |

Fig. 4. Performance comparison with leading optical flow estimation methods on Middlebury test dataset. (Screenshot of website http://vision.middlebury.edu/flow)

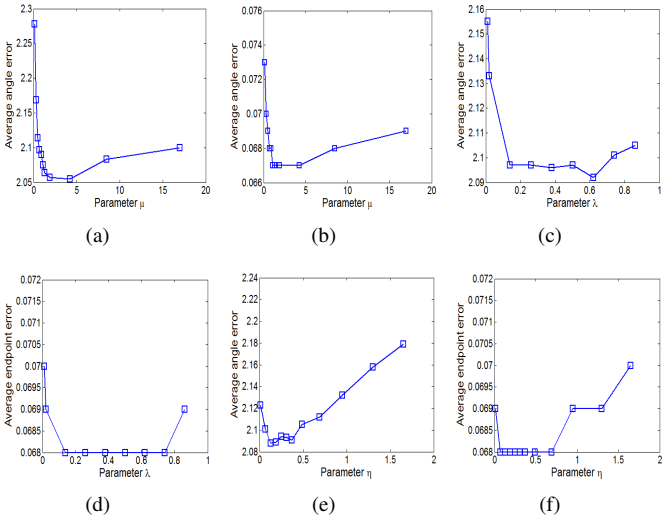



Fig. 6. Average angle error and average endpoint error on *Venus* sequence of Middlebury training set with varying (a) and (b) $\mu$; (c) and (d) $\lambda$; (e) and (f) $\eta$.

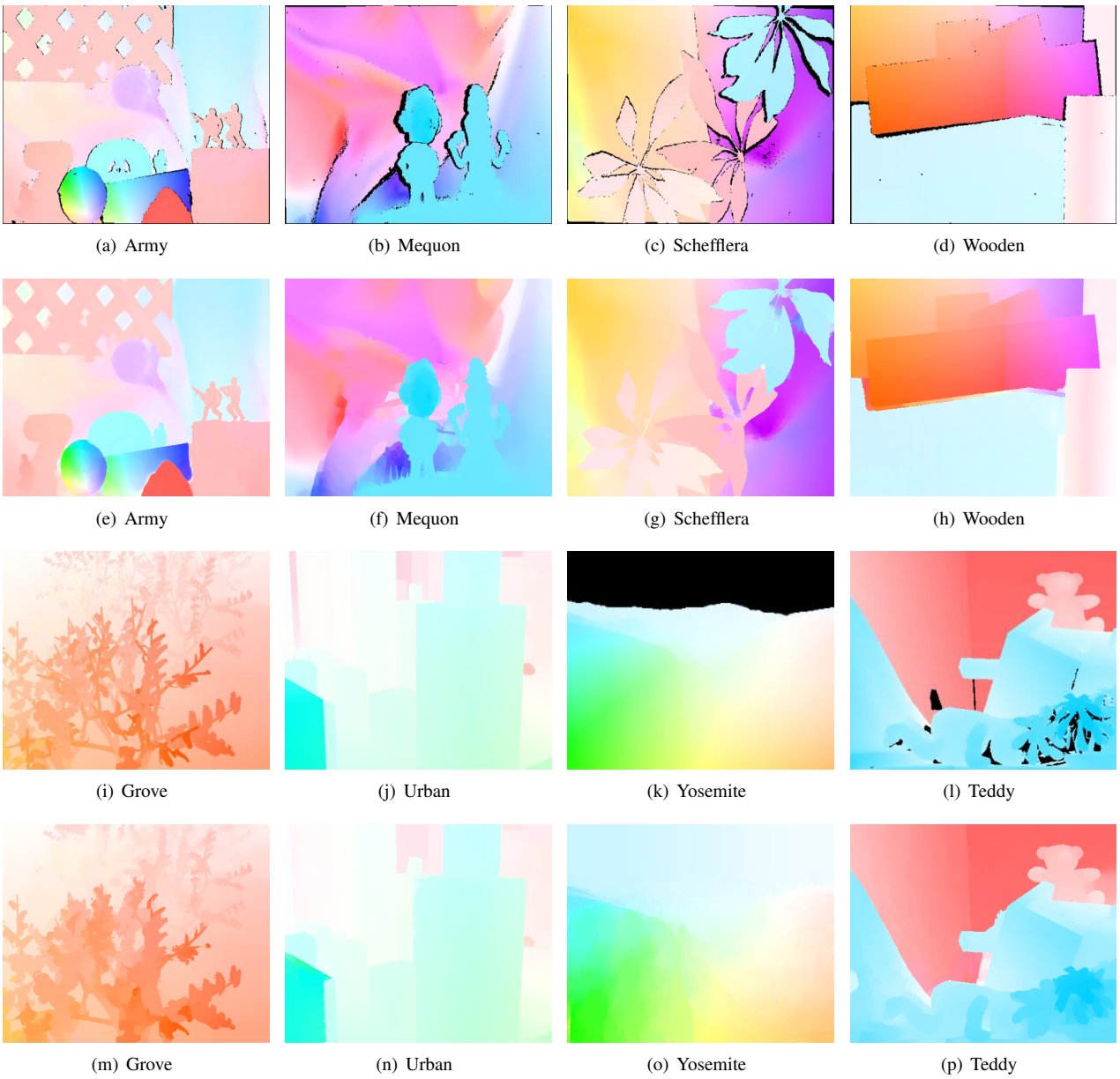
the *Urban* sequence of the Middlebury test set are reported in Table III. By developing a new optimization algorithm to efficiently handle the nonlocal regularization term , the Efficient-NL method [22] is even faster than the Classic+NL method [11]. The running time of the proposed algorithm is generally $8 \sim 10$ times of that of Efficient-NL method [22] due to the high computational complexity of the low-rank and sparse matrix decomposition. When the nuclear norm is used, the running time of the proposed algorithm can be much reduced as less iterations are required for the low-rank and sparse matrix decomposition. As the low-rank and sparse matrix decompositions for each set of similar flow patches can be performed in parallel, the running time of the proposed algorithms can be significantly reduced if the proposed algorithms are implemented with parallel computation technique, and this will remain as the future work.

TABLE III
RUNNING TIME (SEC) ON THE *Urban* DATASET OF MIDDLEBURY BENCHMARK (EXPERIMENTS PERFORMED ON INTEL CORE I7-3770 CPU).

| Classic+NL | LSM | Efficient-NL | SR-LR-NN | SR-LRS-Logdet |
|---|---|---|---|---|
| 225.1 | 1615 | 183.4 | 1476.7 | 1904.5 |

## VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an effective joint low-rank and sparse decomposition model for optical flow estimation. By grouping similar flow patches across a large neighborhood, we regularize the flow field effectively by decomposing the formed matrix into a low-rank component and a sparse component. The low-rank matrix represents the common similar motion pattern, while the sparse component represents the outliers. To better approximate the rank minimization problem and thus further improve the flow estimation performance, we use the $\log\det(\cdot)$ function as a non-convex surrogate



(a) Army (b) Mequon (c) Schefflera (d) Wooden

(e) Army (f) Mequon (g) Schefflera (h) Wooden

(i) Grove (j) Urban (k) Yosemite (l) Teddy

(m) Grove (n) Urban (o) Yosemite (p) Teddy

Fig. 5. Color coded flow results of Middlebury test set. The odd rows show the ground truth from the Middlebury website; the even rows show the estimated flow results by the proposed method.

of the rank. The resulting energy minimization problem is efficiently solved by an alternatively minimization algorithm. Experimental results on the Middlebury dataset demonstrate that the proposed method is very competitive to the current leading nonlocal and sparsity based regularization techniques.

In this paper we mainly to show the effectiveness of the low-rank and sparse decomposition for flow estimation. The performance could be further improved by addressing other complementary issues, such as the depth order and large displacement motion issues. Another possible way to further improve the proposed method is to adopt a more robust flow patch grouping method. Currently, we simply do the patch matching using the color images, without considering the occlusion issue and thus is not yet optimal. We will address these issues in future works.

## ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions that have significantly improved the presentation of this paper.

## APPENDIX A
## ITERATIVE REWEIGHTED LEAST SQUARE (IRLS) SOLVER

The main idea of IRLS is to compute $d\boldsymbol{u}_x$ and $d\boldsymbol{u}_y$ by taking $[\partial E/\partial d\boldsymbol{u}_x; \partial E/\partial d\boldsymbol{u}_y] = 0$, where $E$ denotes the right

hand side of Eq.(21). We derive

$$\begin{aligned}\frac{\partial E}{\partial d\boldsymbol{u}_x} =&\sum_{\boldsymbol{x}} \phi'_D(f_{\boldsymbol{x}})(I_x^2 d\boldsymbol{u}_x + I_x(I_t + I_y d\boldsymbol{u}_y))+\\ &\eta\sum_{\boldsymbol{x}} \phi'_S(g_{\boldsymbol{x}})(\boldsymbol{D}_x^\top \boldsymbol{D}_x + \boldsymbol{D}_y^\top \boldsymbol{D}_y)(d\boldsymbol{u}_x + \boldsymbol{u}^0)+\\ &\frac{\mu}{2}\sum_i \tilde{\boldsymbol{R}}_i^\top(\tilde{\boldsymbol{R}}_i(d\boldsymbol{u}_x + \boldsymbol{u}_x^0) - \boldsymbol{L}_{x,i} - \boldsymbol{S}_{x,i}),\end{aligned} \tag{26}$$

where $f_{\boldsymbol{x}}$ and $g_{\boldsymbol{x}}$ are defined as $f_{\boldsymbol{x}} = |I_t + I_x d\boldsymbol{u}_x + I_y d\boldsymbol{u}_y|^2$ and $g_{\boldsymbol{x}} = |\nabla \boldsymbol{u}_x(\boldsymbol{x})|^2 + |\nabla \boldsymbol{u}_y(\boldsymbol{x})|^2$, respectively. Let $\boldsymbol{I}_x = diag(I_x)$, $\boldsymbol{I}_y = diag(I_y)$, $\boldsymbol{I}_x^2$, $\boldsymbol{\Psi}' = diag(\phi'_D(f_{\boldsymbol{x}}))$, and $\boldsymbol{\Phi}' = diag(\phi'_S(g_{\boldsymbol{x}}))$ be the diagonal matrixes. Then Eq.(26) can be rewritten in matrix form:

$$\begin{aligned}\frac{\partial E}{\partial d\boldsymbol{u}_x} =&(\boldsymbol{\Psi}'\boldsymbol{I}_x^2 + \eta\boldsymbol{L} + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top\tilde{\boldsymbol{R}}_i)d\boldsymbol{u}_x + \boldsymbol{\Psi}'\boldsymbol{I}_x\boldsymbol{I}_y d\boldsymbol{u}_y+\\ &\boldsymbol{\Psi}'\boldsymbol{I}_x\boldsymbol{I}_t + \eta\boldsymbol{L}\boldsymbol{u}_x^0 + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top(\tilde{\boldsymbol{R}}_i\boldsymbol{u}_x^0 - \boldsymbol{L}_{x,i} - \boldsymbol{S}_{x,i}),\end{aligned} \tag{27}$$

where $\boldsymbol{L}$ is defined as

$$\boldsymbol{L} = \boldsymbol{D}_x^\top\boldsymbol{\Phi}'\boldsymbol{D}_x + \boldsymbol{D}_y^\top\boldsymbol{\Phi}'\boldsymbol{D}_y. \tag{28}$$

Similarly, we can also derive

$$\begin{aligned}\frac{\partial E}{\partial d\boldsymbol{u}_y} =&(\boldsymbol{\Psi}'\boldsymbol{I}_y^2 + \eta\boldsymbol{L} + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top\tilde{\boldsymbol{R}}_i)d\boldsymbol{u}_y + \boldsymbol{\Psi}'\boldsymbol{I}_x\boldsymbol{I}_y d\boldsymbol{u}_x+\\ &\boldsymbol{\Psi}'\boldsymbol{I}_y\boldsymbol{I}_t + \eta\boldsymbol{L}\boldsymbol{u}_y^0 + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top(\tilde{\boldsymbol{R}}_i\boldsymbol{u}_y^0 - \boldsymbol{L}_{y,i} - \boldsymbol{S}_{y,i}).\end{aligned} \tag{29}$$

Since both $\frac{\partial E}{\partial d\boldsymbol{u}_x}$ and $\frac{\partial E}{\partial d\boldsymbol{u}_y}$ contain nonlinear functions $\phi'_D(f_{\boldsymbol{x}})$ and $\phi'_S(g_{tx})$, $[\frac{\partial E}{\partial d\boldsymbol{u}_x}; \frac{\partial E}{\partial d\boldsymbol{u}_y}] = 0$ can be solved by the following fixed-point iterations:

**Algorithm 2** IRLS for solving Eq.(21)

(1) Initialize $d\boldsymbol{u}_x = 0$, $d\boldsymbol{u}_y = 0$;
(2) Compute the diagonal matrix $\boldsymbol{\Psi}'$ and $\boldsymbol{\Phi}'$ using the current estimate $d\boldsymbol{u}_x$ and $d\boldsymbol{u}_y$;
(3) Solve the following linear equation:

$$\begin{bmatrix}\boldsymbol{A}_{xx} & \boldsymbol{A}_{xy}\\ \boldsymbol{A}_{xy} & \boldsymbol{A}_{yy}\end{bmatrix}\begin{bmatrix}d\boldsymbol{u}_x\\ d\boldsymbol{u}_y\end{bmatrix} = -\begin{bmatrix}\boldsymbol{b}_x\\ \boldsymbol{b}_y\end{bmatrix} \tag{30}$$

(4) Stop if $d\boldsymbol{u}_x$ and $d\boldsymbol{u}_y$ converge; otherwise, go to (2).

In Eq.(30), $\boldsymbol{A}_{xx}$, $\boldsymbol{A}_{xy}$, $\boldsymbol{A}_{yy}$, $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ are defined as

$$\begin{aligned}\boldsymbol{A}_{xx} =&(\boldsymbol{\Psi}'\boldsymbol{I}_x^2 + \eta\boldsymbol{L} + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top\tilde{\boldsymbol{R}}_i)\\ \boldsymbol{A}_{yy} =&(\boldsymbol{\Psi}'\boldsymbol{I}_y^2 + \eta\boldsymbol{L} + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top\tilde{\boldsymbol{R}}_i)\\ \boldsymbol{A}_{xy} =&\boldsymbol{\Psi}'\boldsymbol{I}_x\boldsymbol{I}_y\\ \boldsymbol{b}_x =&\boldsymbol{\Psi}'\boldsymbol{I}_x\boldsymbol{I}_t + \eta\boldsymbol{L}\boldsymbol{u}_x^0 + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top(\tilde{\boldsymbol{R}}_i\boldsymbol{u}_x^0 - \boldsymbol{L}_{x,i} - \boldsymbol{S}_{x,i})\\ \boldsymbol{b}_y =&\boldsymbol{\Psi}'\boldsymbol{I}_y\boldsymbol{I}_t + \eta\boldsymbol{L}\boldsymbol{u}_y^0 + \mu\sum_i \tilde{\boldsymbol{R}}_i^\top(\tilde{\boldsymbol{R}}_i\boldsymbol{u}_y^0 - \boldsymbol{L}_{y,i} - \boldsymbol{S}_{y,i})\end{aligned} \tag{31}$$

In the above algorithm, we iteratively update the weighting matrixes $\boldsymbol{\Psi}'$ and $\boldsymbol{\Phi}'$ based on the current estimates of $d\boldsymbol{u}_x$ and $d\boldsymbol{u}_y$, and solve the linear equation using the conjugate gradient algorithm.

## REFERENCES

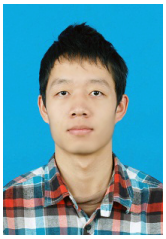

[1] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
[2] M. J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piecewise smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, Jan. 1996.
[3] E. Memin and P. Perez, "Dense estimation and object-based segmentation of the optical flow with robust techniques," *IEEE Trans. Image Processing*, vol. 7, no. 5, pp. 703–719, May 1998.
[4] A. Wedel, D. Cremers, T. Pock, and H. Bischof, "Structure and motion-adaptive regularization for high accuracy optic flow," in *Proc. of ICCV*, 2009.
[5] D. Sun, S. Roth, and M. Black, "Secrets of optical flow estimation and their principles," in *Proc. of CVPR*, 2010.
[6] S. Roth and M. Black, "On the spatial statistics of optical flow," in *Proc. of ICCV*, 2005.
[7] D. Sun, S. Roth, J. P. Lewis, and M. Black, "Learning optical flow," in *Proc. of ECCV*, 2008.
[8] X. Zhang, M. Burger, X. Bresson, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 253–276, 2010.
[9] K. Lee, D. Kwon, I. Yun, and S. Lee, "Optical flow estimation with adaptive convolution kernel prior on discrete framework," in *Proc. of CVPR*, 2010.
[10] M. Werlberger, T. Pock, and H. Bischof, "Motion estimation with non-local total variation regularization," in *Proc. of CVPR*, 2010.
[11] X. Shen and Y. Wu, "Sparsity model for robust optical flow estimation at motion discontinuities," in *Proc. of CVPR*, 2010.
[12] K. Jia, X. Wang, and X. Tang, "Optical flow estimation using learned sparse model," in *Proc. of ICCV*, 2011.
[13] Z. Chen, J. Wang, and W. Ying, "Decomposing and regularizing sparse/non-sparse components for motion field estimation," in *Proc. of CVPR*, 2012.
[14] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. of NIPS*, 2009.
[15] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis," *Journal of the ACM*, vol. 58, no. 3, 2011.
[16] E. Candes and Y. Plan, "Matrix completion with noise," *Proceeding of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.
[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. of ICCV*, 2009.
[19] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. of CVPR*, 2011.
[20] B. Goldluecke, E. Strekalovskiy, and D. Cremers, "The natural vectorial total variation which arises from geometric measure theory," *SIAM J. Imaging Sci.*, vol. 5, no. 2, pp. 537–563, 2012.
[21] G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, "An epigraphical convex optimization approach for multicomponent image restoration using non-local structure tensor," in *Proc. of ICASSP*, 2013, pp. 1359–1363.
[22] P. Krahenbuhl and V. Koltun, "Efficient nonlocal regularization for optical flow," in *Proc. of ECCV*, 2012.
[23] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. of ECCV*, 2004, pp. 25–36.
[24] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, March.
[25] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, 2012.
[26] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006.
[27] J. Cai, E. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, 2010.

[28] M. Fazel, H. Hindi, and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," in *Proc. of Am. Control Conf.*, 2003, pp. 2156–2162.
[29] E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted $l_1$ minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.
[30] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *IJCV*, vol. 92, no. 1, pp. 1–31, 2011.
[31] A. Wedel, T. Pock, C. Zach, D. Cremers, and H. Bischof, "An improved algorithm for tv-l1 optical flow," in *Proc. of Dagstuhl motion workshop*, 2008.
[32] F. Heitz, P. Perez, and P. Bouthemy, "Multiscale minimization of global energy functions in some visual recovery problems," *CVGIP: Image understanding*, vol. 59, no. 1, pp. 125–134, 1994.
[33] D. Sun, E. Sudderth, and M. Black, "Layered segmentation and optical flow estimation over time," in *Proc. of CVPR*, 2012.
[34] ——, "Layered image motion with explicit occlusions, temporal consistency and depth ordering," in *Proc. of NIPS*, 2010.

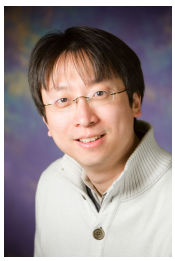
**Weisheng Dong** Weisheng Dong received the B.S. degree in electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004, and the Ph.D. degree in circuits and system from Xidian University, Xian, China, in 2010. He was a Visiting Student with Microsoft Research Asia, Bejing, China, in 2006. From 2009 to 2010, he was a Research Assistant with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. In 2010, he joined the School of Electronic Engineering, Xidian University, as a Lecturer, where he has been an Associate Professor since 2012. His research interests include inverse problems in image processing, sparse signal representation, and image compression. He was a recipient of the Best Paper Award at the SPIE Visual Communication and Image Processing (VCIP) in 2010.

**Guangming Shi** Guangming Shi (SM'10) received the B.S. degree in Automatic Control in 1985, the M.S. degree in Computer Control and Ph.D. degree in Electronic Information Technology, all from Xidian University in 1988 and 2002, respectively.

He joined the School of Electronic Engineering, Xidian University, in 1988. From 1994 to 1996, as a Research Assistant, he cooperated with the Department of Electronic Engineering at the University of Hong Kong. Since 2003, he has been a Professor in the School of Electronic Engineering at Xidian University, and in 2004 the head of National Instruction Base of Electrician and Electronic (NIBEE). From June to December in 2004, he had studied in the Department of Electronic Engineering at University of Illinois at Urbana-Champaign (UIUC). Presently, he is the Deputy Director of the School of Electronic Engineering, Xidian University, and the academic leader in the subject of Circuits and Systems. His research interests include compressed sensing, theory and design of multirate filter banks, image denoising, low-bit-rate image/video coding and implementation of algorithms for intelligent signal processing (using DSP and FPGA). He has authored or co-authored over 60 research papers.



**Xiaocheng Hu** Xiaocheng Hu received his B.S. degree in 2010 from University of Science and Technology of China, where he is now pursuing the Ph.D. degree in University of Science and Technology of China. His research interests include multimedia security, image and video processing, video compression and information hiding.



**Yi Ma** Yi Ma (F'13) is a professor at the School of Information Science and Technology of ShanghaiTech University. He received his Bachelors' degree in Automation and Applied Mathematics from Tsinghua University, China in 1995. He received his M.S. degree in EECS in 1997, M.A. degree in Mathematics in 2000, and his PhD degree in EECS in 2000 all from the University of California at Berkeley. From 2000 to 2011, he was an associate professor of the ECE Department of the University of Illinois at Urbana-Champaign, where he now holds an adjunct professor position. From 2009 to early 2014, he was a principal researcher and the manager of the Visual Computing Group of Microsoft Research Asia. His main research areas are in computer vision and high-dimensional data analysis. He is the first author of the textbook "*An Invitation to 3D Vision*" published by Springer in 2004. Yi Ma was the recipient of the David Marr Best Paper Prize from ICCV 1999 and Honorable Mention for the Longuet-Higgins Best Paper Award from ECCV 2004. He received the CAREER Award from the National Science Foundation in 2004 and the Young Investigator Program Award from the Office of Naval Research in 2005. Yi Ma has been an associate editor for International Journal on Computer Vision, SIAM Journal on Imaging Science, IEEE Trans. on Pattern Analysis and Machine Intelligence, and IEEE Trans. on Information Theory. He served as the Program Chair for ICCV 2013 and is the General Chair for ICCV 2015.