Region Tree based Sparse Model for Optical Flow Estimation

Wei Luo, Fanglong Zhang, Jian Yang and Jingyu Yang
School of Computer Science and Engineering
Nanjing University of Science and Technology
Nanjing, China, 210094
Email: {cswluo, zhangfanlong}@gmail.com, {csjyang, yangjy}@mail.njust.edu.cn

Abstract—Nonlocal regularization has been verified as an effective way to estimate optical flow. Most work in this line constructs the regularizer by only considering the structure of regular grid-like nonlocal neighborhood, but not explicitly takes advantage of the global structure. In this paper, we propose to construct a superpixel based region tree to explicitly incorporate the global structure information into the regularizer. To make use of this non-regular nonlocal (NRNL) regularizer to obtain region-wise smooth and discontinuity preserving flow filed, we first reconstruct the flow for each superpixel by sparse representation, and then dynamically select the superpixel flow with the lowest energy as the optimally-recovered flow field, which corresponds to the optimal sub-region tree. Finally, we update the flow alternatively through continuous optimization. Incorporating the superpixel and sparse representation method not only constrains the nonlocal information that comes from homogeneous region, but also removes the intermediate flow field noise. Experiments on the Middlebury benchmark demonstrate the effectiveness of our method.

I. INTRODUCTION

Optical flow (OF) estimation is a fundamental problem in many computer vision applications. It concerns to recover 2D projection of motion patterns in 3D space. In 2010, Sun et al. [1] elucidated reasons for the success of modern OF algorithms, and pointed out median filtering the noise in intermediate flow field is critical to improve the OF accuracy. They also derived a corresponding nonlocal (also termed as higher-order in MRFs) model for robust OF estimation. Since then, most works [2], [3] apply the nonlocal regularization and median filtering to OF estimation. One disadvantage of this method is that it is difficult to refine flow field region-by-region, because it’s a global variational model. Recently [4] developed a sparse representation based OF model, which recovers flow field by linear combining of very parsimonious wavelet basis. This model can unify different treatments for piecewise smooth, discontinuity preserving and various motion patterns, which were implicitly or explicitly implemented in other models [1], [2], [5]. Furthermore, the sparse model has the ability to recover flow patch-by-patch, but it suffers high computational complexity. More importantly, the sparse codes for neighboring flow patches may vary drastically, since the edge and noise may exist in flow field, and an over-complete dictionary is used. Thus, it generally needs an additional postprocessing to stabilize the flow field, which causes extra computational complexity. In this paper, we will explore a way to make use of the advantages of both methods and eliminate their weaknesses.

Grouping homogeneous pixels into superpixel gains increasing attentions in computer vision [6], [7]. In optical flow, pixel based method suffers aperture problem for untextured areas [8]. Over-segmenting image into homogeneous parts hypothesizes the flow in each segment is smooth, thus it can accelerate the flow propagation, and also prevent the flow moving across the edges. Conversely, over-segmentation may cause wrinkle-like effects in the flow field. In order to remove the wrinkle-like effects, a feasible method is grouping segments with similar flows into a bigger superpixel and then refining the flow in the superpixel as a whole. Based on this merging process, an optimal sub-region or superpixel tree can be determined as the one whose energy evaluated on the flow field is the lowest.

The gradient based variational method, however, is not applicable to refine flow field region-by-region (superpixel), which is becoming increasingly important in hierarchical framework [9], since the flow in the coarse level is served as initialization for a bigger non-convex problem in the next fine level [10]. Therefore, in this paper, we develop a technique for refining flow field region-by-region based on sparse representation. We first construct flow patch for each superpixel, making its dimension consistent with the dimension of dictionary basis, and then reconstruct it by sparse representation. To make the constructed flow patch stable and representative, we filter the constructed flow patch by using mean filter before it being reconstructed. Besides robust to noise, this implementation also implicitly extends the regularizer to an even larger non-regular nonlocal (NRNL) region but with homogeneity preserved.

In summary, we take advantage of image structure information to extend the regular grid-like nonlocal (RNL) regularizer to a larger NRNL regularizer with homogeneity preserving. In order to make use of this NRNL regularizer to obtain more accurate flow field and remove its wrinkle-like effects at the same time, we build up a region tree to sparsely reconstruct the flow field region-by-region. Alternatively optimizing the sparsely recovered OF and the variational OF according to the optimal sub-region tree can guarantee the final flow field preserves sharp discontinuity in boundary area and smoothness in homogeneous area.

The reminder of this paper is organized as follows. Section II presents the mostly related work that inspires our work. In Section III, we first detail the process of building region-tree, then introduce our OF model with NRNL regularizer, and finally present the optimization procedure. The experiments are evaluated in Section IV. And we conclude our work in Section...
II. RELATED WORK

Roth and Black [11] first investigated the higher-order regularization in OF field under the FoE model [12], and they employed Student-t distribution as the expert to model flow patches structure. Sun et al. [13] extended this model by replacing Student-t expert with more flexible GSMs expert, and also learned a higher-order data term. Wedel et al. [14] introduced median filtering technique as an implicit higher-order regularization, but Sun et al. [1] explicitly and thoroughly studied the effects of median filtering for flow field, and elicited their nonlocal model for accurate OF estimation. As mentioned in the previous section, the regularization used in these methods are all RNL regularizer. And they implicitly made use of structure information by local methods, e.g. bilateral filtering. On the contrary, we explicitly exploit the image structure information to regularize the flow field through superpixels.

Sparse OF model [3], [4] is a most recently proposed model which introduces sparse higher-order prior into flow patches. Shen and Wu [4] first studied the sparse properties for flow field, and recovered the flow field by using regular wavelet dictionary. Jia et al. [3] moved the work forward to learned dictionary and cast it into multi-scale framework to handle large displacement. A slightly different idea to impose sparse prior is error separation [15], which regularizes different parts by different norms. The sparse model is compactness, but it lose the flow patches structure, so image-driven or/and flow-driven method usually be used to refine the flow structure. A more serious problem with sparse method is the sparse codes for similar patches are not consistent which in turn bring noise into flow field.

To achieve approximately constant flow in homogeneous region, superpixel based method is employed to offset the influence caused by noise. Lei and Yang [16] made use of this idea to construct a region tree to compute OF. Their region tree is built in each layer of image pyramid by over-segmentation with different granularity. Then starting from the coarsest segmentation layer to the finest one, the corresponding displacement is solved by dynamic programming, and the results of the coarse layer region tree are used by the finer layer to refine the search range of motion displacements. Although our method is also based on region tree, we estimate the OF from the finest layer to the coarsest layer by sparse representation method, which is totally different from their discrete search method.

III. MODEL

Optical flow hypothesizes each pixel intensity is almost the same between two consecutive image frames under small motions in infinitesimal time, then the intensity residual between corresponding pixels $\Delta I = I_0(x, y) - I_1(x + u, y + v)$ can be linear approximated by Taylor expansion, $\Delta I = I_x u + I_y v + I_t$. This model has two unknowns $(u, v)$, but with only one constraint. Gradient smooth term is then added to regularize the flow field since Horn & Schunck [8]

$$E(u, v) = \int_{\Omega} \phi_D(|I_x u + I_y v + I_t|^2) + \lambda \phi_S(|\nabla u|^2 + |\nabla v|^2) d\Omega$$

(1)

where $\phi_D(s^2) = s^2$ is used in [8], and the most popular form in today is $\phi_D(s^2) = \sqrt{s^2 + \epsilon^2}$, $\epsilon = 0.001$, which has the power to preserve sharp edges [10]. This model only utilize the first order information to regularize flow field, but lose the higher-order structure information of the flow field. In the following, we first describe how to build a region tree by exploiting nonlocal neighbourhood structure information in Section III-A (this process is independent to OF), then present how to determine the optimal sub-region tree and introduce our NRNL regularized OF model in Section III-B, and finally detail our optimization procedure in Section III-C.

A. Region Tree

Given an image $I_{h \times w}$, we first segment it into small non-overlapped segments through off-the-shelf segmentation algorithms. In order to make use of these segments to regularize flow field, we need the segments coherent with object
Algorithm 1 Constructing region tree
1. Initialize the merging threshold $E_t = 5$.
2. Segmentation
   Partition $I$ into segments, $r_i, i=1\ldots K$, by using mean-shift algorithm. For each $r_i$, find its neighbors $r_j \in \mathcal{N}_r$, and compute $E_{ij}$ using equation (2).
   Compute the total energy $E_t = \sum E_{ij}$ in the current layer.
3. Merging
   Arrange $E_{ij}$ in a matrix $M_{K \times K}$, find the minimum energy in each column $E_{mj}, j=1\ldots K$.
   Checking $M_{j,m}$ whether is the position with the minimum energy in column $m$.
   If true and $E_{mj} < E_c$ and $E_{jm} < E_c$.
   Merge the two segments $r_m$ and $r_j$ into a big superpixel, update its $f, m, c$ and neighbors, Compute the merged total energy $E_{ml}$ as step 2.
   If $E_{ml} < E_t$, set $E_t = E_{ml}$ and update $M_{K \times K}$.
4. Back to step 3 until no new superpixel comes up or reaching a preset number of iterations.

boundary if there really exists boundary. For this reason, we employ an improved variant of the mean-shift method [17] to generate segments for our region tree. Without loss of generality, supposing the image $I$ has been over segmented into $K$ small segments $r_i, i=1\ldots K$, and each segment $r_i$ connects to its neighbours $r_j \in \mathcal{N}_r$, through edge $c_{ij}$. Then, the feature $f$ for each segment can be computed, e.g. SIFT or LBP [18], and the energy associated with two neighboring segments, $r_i$ and $r_j$, can be defined as:

$$E_{ij} = \exp^{-\kappa c_i \exp\left(\frac{E(c_i, c_j)}{\sigma_1^2} + 0.5 \frac{D(m_i, m_j)}{\sigma_2^2} + 0.5 \frac{KL(f_i|f_j)}{\sigma_3^2}\right)}$$

$$D(c_i, c_j) = \begin{cases} D(c_i, c_j), & \text{if } D(c_i, c_j) \leq \log(h^2 + w^2) \\ 20, & \text{otherwise} \end{cases} \quad (2)$$

where $KL(p|q)$ is K-L divergence, $m$ and $\kappa$ are mode feature and boundary strength computed by mean shift algorithm, $c$ represents the central coordinate of segment and $D$ measures the Euclidean distance between two segments. Here, we adopt weighted K-L and $D$ to balance the influence of each element in the exponential term. Comparing to [16], our energy formulation considers more detail property of flow field, such as feature $f$ (LBP feature is used in our experiments), which is especially useful when dealing with large displacement in hierarchical framework, since the importance of features varies with the image resolution changes. The segments were generated based on image characteristic, e.g. texture and color, which can be considered as an image driven approach when using them to regularize flow field. What’s more important is that due to the segments preserve the structure of image, they can constrain the structure of flow field. Additionally, different initialization for the segmentation algorithm may result in different segments, thus affect the final OF estimation, we leave this discussion in our feature work, and focus on the study of the feasibility of this global regularization method here. The algorithm for constructing region tree is presented in Algorithm 1. And we also show the process of merging and refining OF in Figure 1.

B. Reconstruct Flow Field Region-by-Region through Sparse Representation

Supposing we already have an initialized flow field $u$, we then build an optimal sub-region tree according to the energy of sparse reconstructed flow field as follows: 1) reconstruct flow for each node (superpixel) of the tree; 2) for each parent node, use its flow to replace its children flow if this operation can decrease the energy according to equation (1); 3) repeat this procedure until reaching the root node. Following this procedure, our OF model is defined as: (For clarity, we simplify $u = (u, v)^T$, $\nabla u = (\nabla u, \nabla v)^T$ and $\nabla I = (I_x, I_y)^T$

$$E(u, \tilde{u}, \alpha) = \sum_{j \in I} \phi_D(\nabla T^2 u + \nabla I_t^2) + \lambda_1 \phi_S(\nabla u^2)$$

$$+ \sum_{r} \left( \lambda_2 \| u - \tilde{u} \|_2 + \lambda_3 (\| u - D\alpha \|_2 + \gamma |\alpha|) \right) \quad (3)$$

where $\Omega = \sum r$ represents the whole image space and $r$ is the index of superpixels which corresponds to nodes of the optimal sub-region tree, $\tilde{u}$ denotes auxiliary flow field, $D$ and $\alpha$ is the dictionary and sparse codes, respectively. The normal term is the same as equation (1), the coupling term constrains the flow deviations between $u$ and $\tilde{u}$, and the sparse term is our NRNL regularizer to constrain flow smoothness in each region $r$. While this model is very similar to [1], the sparse regularization term is completely different.

Different initial flow field and sparse reconstruction configuration will result in different sub-region tree. For this reason, we use the public available Classic+NL codes and its default settings to initialize the flow field in our experiments, and focus on the effectiveness of NRNL regularizer and the difference when complemented the RNL regularizer with the NRNL regularizer.

The noise in intermediate flow field and the dimension difference between segment and dictionary basis need to be systematically handled before working well with this model. Furthermore, the noise can make sparse coding, $\min_\alpha \| \alpha \|_1, s.t. \| u - D\alpha \|_2^2 < \epsilon$, unstable, which is reflected in the code $\alpha$. Thus, to obtain stable and smooth flow in each noisy segment $r_i$, we compute $d$ representative flows for each segment, which are uniformly spaced on each segment. The $d$ is the dimension of the basis in dictionary $D$. We compute $d$ representative flows by first dividing the segment into $d$ uniform parts and then computing the flow’s mean value in each part. To filter out the noise and exploit the effectiveness of NRNL regularizer, we instead average filtering the $d$ flows in each segment to output its final $d$ representative flows. Although this construction method for representative flows is intuitive and simple, and the representative accuracy may degenerate with the segment size increasing, it works well in our experiments. The reason maybe the average filtering smooths the discrepancy between these representative flows, while the smooth flow in each segment is exactly what we need for OF estimation.

C. Optimization

We alternatively optimize our OF model (3) to accurately recover the flow field. The penalty function used in our model is $\phi_\epsilon(s^2) = \sqrt{s^2 + \epsilon^2}$, which is approximate to the
\( L1 \) penalty but with continuous differentiable. Because this penalty function is convex and the factors \( I_x, I_y, I_{x^2}, I_{y^2}, \nabla u \) and \( \nabla v \) are all linear function to \( u \) and/or \( v \), we can minimize it in each alternative step as [1] by setting
\[
\nabla_u E(u, v) = 0 \quad \text{and} \quad \nabla_v E(u, v) = 0 \tag{4}
\]
Linearizing the gradient function is the main difficult to solve equation (4), because the warping step is involved in each step. For this we follow the approach of Brox et al. [10] to use two nested fixed point iteration to linearize the gradient function. We then rearrange the results into a system of sparse linear equation, which can be solved by standard technique. Specifically, our algorithm proceeds with initial \( u = u^0 \) and the following iterations:

- For \( u \) being fixed, update \( \tilde{u} \) by minimizing
  \[
  E(\tilde{u}, \alpha) = \sum_r \lambda_2 \|u - \tilde{u}\|_1 + \lambda_1 \|D\tilde{u} - D\alpha\|_2^2 + \gamma |\alpha|_1
  \]
  via soft-thresholding [19] and sparse coding [20], where \( r \) is the index to nodes of our optimal sub-region tree. We learn dictionary \( D \) and compute \( \alpha \) by using Mairal’s [20] method.

- For \( \tilde{u} \) being fixed, update \( u \) by minimizing
  \[
  E(u) = \sum_{\Omega} \phi_D(|\nabla I|^2 + |\nabla I|^2) + \lambda_1 \phi_S(|\nabla u|^2) + \lambda_2 \|u - \tilde{u}\|_1
  \]
  using equation (4) with gradient descent method.

We initialize \( \tilde{u} = u \) when optimizing \( \tilde{u} \), and then alternatively optimize \( \alpha \) and \( \tilde{u} \). While fixing \( \tilde{u} \) to optimize \( \alpha \) can be done by sparse coding, optimizing \( \tilde{u} \) can employ soft-thresholding method [19]. This formulation can guarantee to find the optimal solution, because \( E(\tilde{u}, \alpha) \) is convex when \( \alpha \) is fixed. After finding the optimal \( \tilde{u} \), it proceeds to optimize \( u \) by using (4). This procedure alternatively proceeds until the energy is converged or the difference between \( E_u \) and \( E_{\tilde{u}} \) is reduced to a preset threshold.

We cast this optimize method into a coarse-to-fine hierarchical framework to refine the flow field which will be re-scaled to larger size to serve as the initialization for the next higher resolution level. Indeed, our method can be cast into every warping step, but building optimal sub-region tree for reconstructing flow after each warping step will be time consuming. We’ll show in the next section, even though we only reconstruct flow after the final warping step in each scale, we can also obtain high accurate results.

IV. EXPERIMENTS

In this section, we quantitatively evaluate our method for two-frame sequences optical flow on the Middlebury dataset, which contains training data with ground truth flow field and test data without ground truth. The training data contains 3 nonrigid scenes (RubberWhale, Hydrangea, Dimentrodon), 4 synthetic scenes (Grove2, Grove3, Urban2, Urban3) and 1 modified stereo rigid scene (Venus). Our method is embedded in a coarse-to-fine hierarchical framework to cope with large displacement. To alleviate the influence of lighting change, we apply the structure-texture decomposition to the original input sequence as [14]. To limit the influence of spurious local optimal, we follow [13] to use graduated non-convexity (GNC) method to optimize \( u \). For efficiency, we only apply our NRNL regularizer once after the final warping step in each level.

The minimal granularity in mean shift algorithm is set to 25 to be consistent with the dimension of the dictionary basis. We construct a 3-layer region tree and set the merging threshold \( E_c = 5 \). The LBP feature is computed and normalized into 32 bins for each node (segment). For sparse reconstruction, we randomly select 500 flow patches with size \( 5 \times 5 \) from each training data, and employ Mairal’s [20] method to learn dictionary. Specifically, we learn dictionary for horizontal flow and vertical flow separately with size \( 25 \times 100 \) and combine them to a 50 × 200 dictionary, \( D \), with the reconstruct precision \( \epsilon = 10^{-3} \). In our implementation, we found 3 iterations for alternatively optimizing \( \tilde{u} \) and \( \alpha \) is enough to converge.

In our first experiment, we verify the effectiveness of our proposed NRNL regularizer for OF estimation. We use 3 stage GNC optimization. In stage 1, we build 5-level pyramid with a downsampling factor 0.5, and 2-level pyramid with a downsampling factor 0.8 for stage 2 and stage 3. 10 warping step is implemented in each level and \( \lambda_1 = 3, \lambda_2 = 0.1, \lambda_3 = 1 \) and \( \gamma = 0.05 \) is used in this experiment. Table I compares the average end point error (AEPE) and average angular error (AAE) of our method with \( TV-L1 \) [14], \( \text{Classic++} \) [1], and \( \text{NL-TV-TNCC} \) [21] methods in Middlebury training data. The \( TV-L1 \) and \( \text{Classic++} \) model implicitly implement nonlocal regularization through median filtering but do not utilize the nonlocal structure information. The \( \text{NL-TV-TNCC} \) explicitly implements nonlocal regularization with structure information but constrains it to regular size. The results for \( TV-L1 \) and \( \text{NL-TV-TNCC} \) comes from the original paper separately.

Our method obtains average AAE of 3.237 and AEPE of 0.276 on the Middlebury training data, the improvement is especially significant in cases where existing large rigid object motion when compared with \( \text{Classic++} \) method, e.g. RubberWhale, Urban2, Urban3 (see Table I). While comparing to \( \text{NL-TV-TNCC} \) method, our method exceeds \( \text{NL-TV-TNCC} \) on nonrigid scenes and \( \text{Venus} \), but the best performance in synthetic data is alternatively appearing in each method according to AEPE. The result signifies two points:

<table>
<thead>
<tr>
<th>TABLE I: Evaluation results (AAE/AEPE) on the Middlebury training data</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV-L1</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>RubberWhale</td>
</tr>
<tr>
<td>Hydrangea</td>
</tr>
<tr>
<td>Dimentrodon</td>
</tr>
<tr>
<td>Grove2</td>
</tr>
<tr>
<td>Grove3</td>
</tr>
<tr>
<td>Urban2</td>
</tr>
<tr>
<td>Urban3</td>
</tr>
<tr>
<td>Venus</td>
</tr>
<tr>
<td>AAE/AEPE</td>
</tr>
</tbody>
</table>

* means unavailable.
1) the NRNL regularizer can effectively improve the flow field accuracy, 2) OF accuracy can also be benefit from the regular nonlocal regularizer with structure information. Although the performance for NL-TV-TNCC and our method is very approximate in some sequences according to AEPE, the realization procedure is very different. The RNL regularization need to be implemented after every warping step, while the NRNL regularization only need to be performed once after the final warping step in each level.

In our second experiment, we demonstrate the enhanced performance can be obtained when complementing the RNL regularizer with NRNL regularizer. Note, the purpose of our work is not to deny the effectiveness of RNL regularizer, but to empirically verify the improvement of OF after including the NRNL regularizer. To this end, we follow the nonlocal model proposed by Sun et al. [1] and set the nonlocal range to $7 \times 7$ and $15 \times 15$ in edge and non-edge area, respectively. The weight for the RNL regularizer is 3. We instead use 2 stage GNC optimization in this experiment. In stage 1, we build 5-level pyramid with a down-sampling factor 0.5, and 2-level pyramid with a down-sampling factor 0.8 for stage 2. 3 warping step is implemented in each level and $\lambda_1 = 3$, $\lambda_2 = 0.1$, $\lambda_3 = 1$ and $\gamma = 0.05$.

We evaluate our results on the Middlebury training data with comparison to two closely related methods, Classic+NL [1] and LSM [3]. The both were explicitly using RNL regularizer and exploiting image structure to refine OF. The Classic+NL method makes use of weighted median filter to filter the intermediate flow field in boundary regions with weights computed based on image structures, and un-weighted median filter were applied to non-boundary regions. The LSM method computes OF pixel-by-pixel, although its high performance, it is very time consuming. In our method, we use sparse representation to recover flow field region-by-region. Note, we implement mean filtering in non-edge flow area after each warping step to reduce the accumulated intermediate noise.

Our method obtains average AAE of 2.624 and AEPE of 0.217 on the Middlebury training data (see Table II), which is better than Classic+NL method and LSM method in average. From Table II, we find the performance of our method overwhelms LSM on almost all sequences according to AEPE except on RubberWhale. And also the performance of our method exceeds Classic+NL on almost all sequences according to AAE except Urban2. The results signifies that complementing the RNL regularizer with NRNL regularizer can benefit the OF estimation in general and also verifies the effectiveness of our method to construct NRNL regularizer.

The using of mean filter to filter intermediate flow field in non-edge area after every warping step is decisive for our performance. While our NRNL is based on superpixel that constrains similar structure in each, it agrees with the OF hypothesis that similar structure areas should preserve smooth flow. And the mean filtering in non-edge area just captures the essential of this hypothesis, that establishes its importance role for our method. This is also what we argued to use mean filtering to construct representative flows for each segment in Section III-B. While the median filtering is critical to denoise the intermediate flow filed, it doesn’t collaborate with its homogeneous neighbours to generate a smooth flow field.

We further compare the recovered flow field from Classic+NL and our method for urban2 sequence, on which the performance of our method is less accurate than Classic+NL according to either AEPE or AAE (see Figure 2). The urban2 sequence contains a large motion range. From comparison, we find our method can recover a very smooth flow field in homogeneous area, which is failed in Classic+NL method. The visual comparison also verifies the advantages of our NRNL regularizer and mean filtering strategy. For completeness, We also show our results on the Middlebury test data in Figure 3.

TABLE II: Evaluation results (AAE/AEPE) on the Middlebury training data

<table>
<thead>
<tr>
<th></th>
<th>LSM</th>
<th>Classic+NL</th>
<th>Ours method</th>
</tr>
</thead>
<tbody>
<tr>
<td>RubberWhale</td>
<td>2.285/0.072</td>
<td>2.401/0.076</td>
<td>2.378/0.075</td>
</tr>
<tr>
<td>Hydrangea</td>
<td>1.803/0.151</td>
<td>1.824/0.151</td>
<td>1.806/0.151</td>
</tr>
<tr>
<td>Dimetrodon</td>
<td>2.541/0.129</td>
<td>2.280/0.117</td>
<td>2.268/0.116</td>
</tr>
<tr>
<td>Grove2</td>
<td>1.511/0.105</td>
<td>1.410/0.098</td>
<td>1.377/0.095</td>
</tr>
<tr>
<td>Grove3</td>
<td>5.005/0.473</td>
<td>4.927/0.464</td>
<td>4.908/0.471</td>
</tr>
<tr>
<td>Urban2</td>
<td>2.004/0.221</td>
<td>2.034/0.210</td>
<td>2.093/0.220</td>
</tr>
<tr>
<td>Urban3</td>
<td>2.599/0.375</td>
<td>3.160/0.421</td>
<td>2.909/0.372</td>
</tr>
<tr>
<td>Venus</td>
<td>3.297/0.235</td>
<td>3.289/0.232</td>
<td>3.256/0.232</td>
</tr>
<tr>
<td>AAE/AEPE</td>
<td>2.631/0.220</td>
<td>2.666/0.221</td>
<td>2.624/0.217</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we propose using NRNL regularizer to estimate OF. The NRNL regularizer extends the regular nonlocal regularizer to a larger non-regular neighborhood, however, with homogeneity preserving. Representative flows based sparsely reconstruct according to the region tree makes our method has the ability to update flow field region-by-region. Experiments on the Middlebury benchmark demonstrate the effectiveness of our method, even if we only applied the NRNL regularizer to flow field one time after the final warping step in each level.
Fig. 3: Recovered flow field for the Middlebury test data.

REFERENCES


