OPTICAL FLOW ESTIMATION USING APPROXIMATE NEAREST NEIGHBOR FIELD FUSION

O. U. Nirmal Jith   *S. Avinash Ramakanth*   *R. Venkatesh Babu*

Video Analytics Lab,
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore, India

ABSTRACT

This paper proposes an optical flow algorithm by adapting Approximate Nearest Neighbor Fields (ANNF) to obtain a pixel level optical flow between image sequence. Patch similarity based coherency is performed to refine the ANNF maps. Further improvement in mapping between the two images are obtained by fusing bidirectional ANNF maps between pair of images. Thus a highly accurate pixel level flow is obtained between the pair of images. Using pyramidal cost optimization, the pixel level optical flow is further optimized to a sub-pixel level. The proposed approach is evaluated on the middlebury dataset and the performance obtained is comparable with the state of the art approaches. Furthermore, the proposed approach can be used to compute large displacement optical flow as evaluated using MPI Sintel dataset.

Index Terms— Optical flow, Approximate Nearest Neighbor Field, PatchMatch.

1. INTRODUCTION

Motion estimation of pixels in sequences of consecutive images is called optical flow (OF) and it is widely utilized in various computer vision algorithms including object detection, object tracking, motion estimation etc. Optical flow computation is a well established problem which was introduced by Horn and Schunck [1]. Though various researchers have contributed to develop accurate OF technique, the problem of fast and accurate OF is still not solved with required level of perfection. A study on Middlebury [2] dataset shows that the real-time estimation of OF using CPUs is still not possible with the existing algorithms. Some of the widely used algorithms to compute optical flow are pyramidal LK optical flow [3], optical flow using descriptor matching by Brox et al. [4] and optical flow by solving pixel labelling problem as proposed by Lu et al. [5]. In this proposed approach we adapt Approximate Nearest Neighbor Field (ANNF) maps to compute the optical flow. The aim of ANNF algorithm is to find the closest patch, in euclidean or any other relevant space in source image, for every patch in target image, for a given pair of source and target images. ANNF mappings find its application in image super resolution [6], optic disk detection [7], image retargetting [8], video segmentation [9] and other allied image manipulation algorithms also. In contrast to this conventional use of ANNF mapping, we use ANNF maps to relate two successive images in an image sequence.

The organization of the paper is as follows: section. 2 gives a brief description of state-of-the-art techniques in ANNF and OF computation. section. 3 describes the proposed algorithm in detail. section. 4 shows the experimental results obtained using the proposed approach in comparison with those obtained from other approaches. The paper is concluded in section. 5.

This work was supported by Joint Advanced Technology Programme (JATP), Indian Institute of Science, Bangalore, India.

2. RELATED WORK

Optical flow algorithm was initially formulated as an energy minimization problem by Horn and Schunck [1]. It was done by optimizing the objective function containing data term and smoothness term. Black and Anandan [10] improved upon this by introducing nonconvex robust penalty functions to reduce the outliers. Sun et al. [11] empirically demonstrated the effectiveness of Charbonnier function as penalty function, in optimizing the energy function. Large displacement flow estimation using descriptor matching was introduced by Brox et al. [4]. Xu et al. [12] proposed a new frame work taking pixel-wise scales into consideration in optical flow estimation. Optical flow using rank transform was introduced by Demetz et al. in [13]. Lu et al. [5] formulated optical flow as a pixel labeling problem and proposed a generic Patch Match Filter (PMF) framework for solving discrete multi-labeling problem. Chen et al. [14] formulated motion estimation as a motion segmentation problem and obtained the initial segmentation using approximate nearest neighbors. Introduction of deep matching in large displacement optical flow was done by Weinzaepfel et al. [15].

PatchMatch [8] is an efficient algorithm for the computation of ANNF maps. It is based on the concept of coherency between the images i.e. if two patches are similar in a pair of images, then their neighboring patches will also be similar. Further improving upon PatchMatch, Korman et al. [16] proposed coherency sensitive hashing, based on locality sensitive hashing functions. In a more recent approach, Ramakanth et al. [17] [18] proposed FeatureMatch (FM), using intelligent feature extraction in conjunction with kd-trees for accurate and efficient ANNF computation. In the present work, we extend the concepts proposed in FM to compute optical flow between two consecutive frames.

3. PROPOSED ALGORITHM

The proposed approach is described in detail in this section. Proposed algorithm consists of three steps: (1) Approximate Nearest Neighbour Field (ANNF) computation using FM (2) improving ANNF mapping to obtain pixel-level accurate optical flow and (3) flow refinement to obtain sub-pixel accurate optical flow.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

Algorithm 1 Optical flow using feature match

Input: Previous frame (I_1), Current frame (I_2).
Output: FinalFlow.
Pseudocode:

$flow_{12} = \mathbf{FM}(I_1, I_2)$
$flow_{21} = \mathbf{FM}(I_2, I_1)$
for $AllPixelsIn(I_1)$ **do**
  $[maps] = \mathbf{FindMapping}(flow_{21}, i)$
  if $Length(maps) > 1$ **then**
    $flow'_{12} = \mathbf{FindBestMap}(maps, i)$
  end if
end for
$PFlow = \mathbf{FindBestMap}(flow'_{12}, flow_{12})$
$FinalFlow = \mathbf{FlowRefinement}(PFlow)$

FM(Target, Source): FM computes ANNF mapping for every patch in target image to source image.
FindMapping(flow, i): Function finds all the mappings that maps to i^{th} pixel in I_1.
FindBestMap(flow, i): Function finds the best map to i^{th} pixel in I_1.
FindBestMap($flow_1$, $flow_2$): Function finds the best map from flows $flow_1$ and $flow_2$ using Charbonnier penlty function as explained in section 3.1.
FlowRefinement(pixellevelflow): Refine the flow $PFlow$ using flow refinement as explained in section 3.3.

3.1. ANNF computation

The first step of the proposed algorithm is the computation of ANNF using FM with the given sequence of images. To compute ANNF mapping, FM algorithm [17] uses features extracted from $p \times p \times 3$ (*ie.* a $p \times p$ patch in RGB image) patches of the target and source frames to compute the ANNF map. The features used in FM are as follows:

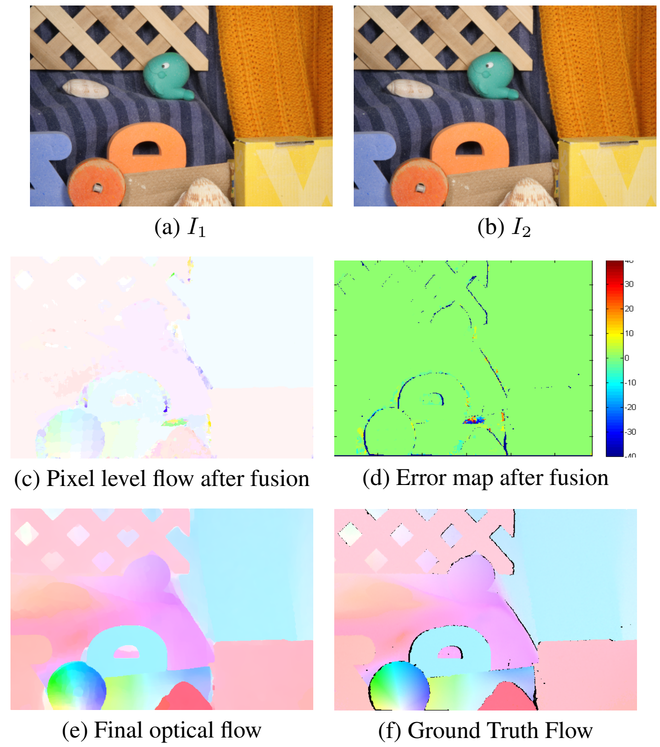
- The mean of R, G, B color channels of the patch.
- The mean of x and y gradient of the patch.
- The first two frequency components of Walsh-Hadamard basis (WH1, WH2) [19].
- The maximum value of the patch.

In addition to these FM features, x and y coordinates are also included for the initialization of optical flow algorithm. Since the two successive frames are closely related, the additional coordinate information improves the ANNF computation. These features are chosen because, they can be computed efficiently using integral images. These features were shown to be computationally efficient compared to other dimension reduction techniques [17].

The initial ANNF map is noisy, and it is given in the literature [17] that coherency between pair of images is an important feature which determines the accuracy of ANNF maps. Coherency means, if two patches from pair of images are similar then their neighboring patches will also be similar. This problem can be ameliorated by including a coherency stage after ANNF computation. Coherency stage does this by selecting the flow that minimizes the error between the two patches from consecutive frames using a Lorentzian penalty function $log(1 + \frac{x^2}{\sigma^2})$ [10].

3.2. Flow fusion

As discussed in section 3.1, ANNF map in both directions between the two images I_1 (previous frame) and I_2 (current frame), as shown in the Fig. 1(a) and 1(b), will differ depending on the choice of target and source images from the sequence of images. These bi-directional flows are obtained by finding the nearest neighbours for target image patches from the source image patches. The flow from I_1 to I_2 ($annf_{12}$) is obtained by making the image frame I_1 as target. In the same way, the flow from frame I_2 to I_1 ($annf_{21}$) is



(a) I_1 (b) I_2
(c) Pixel level flow after fusion (d) Error map after fusion
(e) Final optical flow (f) Ground Truth Flow

Fig. 1: Various Stages of proposed approach

obtained by making I_2 as the target frame. The novelty of our algorithm is to obtain the best mapping from I_1 to I_2, by utilizing $annf_{21}$ and $annf_{12}$. Let us consider the scenario in which the pixel a_1 is mapped to pixel a_2, obtained from $annf_{12}$. And pixel p_2 is mapped to pixel a_1, obtained from $annf_{21}$. Here, from $annf_{21}$ mapping we are getting additional information regarding the pixel a_1, that it is matching to pixel p_2 as well. This way, along with $annf_{12}$, $annf_{21}$ gives information regarding the matching pair for all the pixels in I_1. This additional information helps in filtering the noisy $annf_{12}$ mapping.

The initial flow using ANNF map which are similar in both $annf_{12}$ and $annf_{21}$ can be considered as the true mapping from frame I_1 to I_2. There may be cases when these are not similar and $annf_{21}$ has got more than one mapping to a patch in I_1. In such cases, the best map among the multiple mappings are found using Charbonnier penalty function $\rho(x) = \sqrt{x^2 + \epsilon^2}$ [20]. Thus a pixel in I_1 has got two candidate mappings, one is the map obtained from $annf_{12}$ and another is the best map obtained from the multiple mappings in $annf_{21}$. The final mapping from these two mappings for the pixel is obtained by minimizing Charbonnier penalty function .

3.3. Flow refinement

Estimation of pixel level flow between the images using FM are described in previous sections. This section gives the description about the steps involved in obtaining sub-pixel level accurate optical flow. In order to obtain that, further refinement of the pixel level flow is required. This is done through a refinement stage using the improvised optical flow model introduced by Sun et al.[11, 21].

The Eq. (1) is the improved objective function by Sun et al. [11] which is used for the estimation of optical flow. It has got two parts,

	RubberWhale		Dimetrodon		Hydrangea		Grove3		Urban3		Venus		Grove2		Urban2	
	AAE	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**
annf12	2.25	0.738	2.427	0.124	1.863	0.153	4.90	0.538	12.26	0.829	3.21	0.235	1.398	0.967	2.48	0.45
fusion	2.25	0.737	0.242	0.124	1.862	0.153	4.79	0.498	11.00	0.99	3.168	0.234	1.396	0.968	2.31	0.443

Table 1: Error comparison for different initialization strategies

	RubberWhale		Dimetrodon		Hydrangea		Grove3		Urban3		Venus		Grove2		Urban2	
	AAE	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**	**AAE**	**AEE**
Lorentzian	2.25	0.737	0.242	0.124	1.862	0.153	4.79	0.498	11.00	0.99	3.168	0.234	1.396	0.968	2.31	0.443
Charbonnier	2.25	0.738	0.242	0.124	1.864	0.153	4.77	0.498	11.107	1.109	3.197	0.235	1.40	0.972	2.53	0.633
Squared norm	2.23	0.737	0.242	0.124	1.862	1.535	5.22	0.576	11.008	0.99	3.168	0.234	1.39	0.968	2.55	0.635
without coherency	2.25	0.738	0.242	0.124	1.86	0.153	4.86	4.96	18.743	2.69	3.200	0.23	1.39	0.96	9.87	0.403

Table 2: Error comparison for different cost functions in coherency stage

the one inside the curly braces is the data term and the other is the coupling term. The optimization of objective function is done by treating each term separately.

$$
\begin{aligned}
E(u,v,\hat{u},\hat{v}) &= \sum_{i,j}\{\rho_d(I_1(i,j) - I_2(i+u_{i.j}, j+v_{i,j})) \\
&+\lambda_1[\rho_s(u_{i,j}-u_{i+1,j}) + \rho_s(u_{i,j}-u_{i,j+1}) \\
&+\rho_s(v_{i,j}-v_{i+1,j}) + \rho_s(v_{i,j}-v_{i,j+1})]\} \\
&+\lambda_2(\|u-\hat{u}\|^2 + \|v-\hat{v}\|^2) \\
&+\sum_{i,j}\sum_{(i',j')\in N(i,j)} \lambda_3(|\hat{u}_{i,j}-\hat{u}_{i',j'}| + |\hat{v}_{i,j}-\hat{v}_{i',j'}|)
\end{aligned}
\tag{1}
$$

Here u and v are the horizontal and vertical components respectively of the optical flow; $\lambda_1, \lambda_2, \lambda_3$ are the regularization parameters, ρ_d and ρ_s are the penalty functions respectively for the data and flow terms. The formal connection between the coupling term and median filtering is provided by Li and Osher [22]. Hence, solving Eq. (1) includes, obtaining the flow that optimizes the data term and performing a weighted median on the obtained flow. The detailed explanation about the optimization is given in the literature on secrets of optical flow by Sun et al. [11, 21]

Figure (1) shows the output from different stages of the proposed algorithm. Figures 1(a) and 1(b) are the two consecutive frames of the sequence. Flow obtained using the fusion of ANNF mapping is shown in the Fig. 1(c). Figure 1(d) shows the pixel level accuracy compared to the ground truth flow. Experimentally it is observed that, we are able to get 90% pixel level accuracy at this stage of algorithm. Figure 1(e) gives the final optical flow using the proposed algorithm and Fig. 1(f) shows the ground truth. The pseudo-code for the proposed algorithm is given in **Algorithm 1**.

4. EXPERIMENTATION AND RESULTS

The proposed optical flow algorithm was evaluated using the Middlebury flow benchmark [23]. The algorithm is implemented using MATLAB, on Intel i7 CPU with 3.4 GHz processor and 8GB RAM. With the current CPU implementation, the whole program takes 215 sec running time to compute a sub-pixel flow field for an image pair with resolution 640×480, for instance, the $Urban$ sequence. FM algorithm takes patches with a size ($p \times p$) of 8×8, to extract 10 D features from target and source images. Lorentzian function sigma values, $\sigma_d = 1.5$ and $\sigma_s = 0.03$ are chosen for the data term and spatial term in Eq. 1, in the coherency stage.

4.1. Quantitative Evaluation

In this section, we discuss the various experiments conducted and the comparison of proposed approach with other optical flow algorithms. Average angle error (AAE) and average end point error (AEE) are the two error measures used for the evaluation of dataset. The following Eq. (2), (3) are used for computing these errors:

$$AE = \cos^{-1}\left(\frac{1.0 + u * u_{GT} + v * v_{GT}}{\sqrt{1.0+u^2+v^2}\sqrt{1.0+u_{GT}^2+v_{GT}^2}}\right) \tag{2}$$

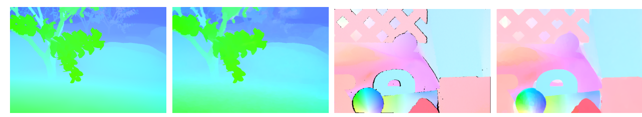
$$EE = \sqrt{(u-u_{GT})^2 + (v-v_{GT})^2} \tag{3}$$

where (u, v) is the flow obtained using the proposed algorithm and (u_{GT}, v_{GT}) is the ground truth flow. Apart from two error measures, other qualitative measurements for the test data are available in evaluation section of middlebury website [23].

Table (1) shows the error values for the final optical flow on Middlebury training dataset with different ANNF initialization. For the first experiment, ANNF initialization is done using the mapping from first image to second image only. For the second experiment, mappings from first to second and vice versa, are considered. Among these, the best flow is taken as discussed in section 3.2. It can be inferred from the table that the error is less in the case of initialization with ANNF fusion, especially in the case of sequences like $Urban3$ and $Grove3$, which have large motion. The error variation can be observed from table (1).

Table (2) shows the effect of using different cost function in the coherency stage. It is clear from the table that the error values are almost similar in all the three cases. Only in Urban sequence we can see a significant variation in AAE and AEE. If we take the average error considering all the training dataset, Lorentzian outperforms all the other cost functions. Hence we have used Lorentzian as the cost function for the evaluation of the dataset.

Table (3) shows the comparison of the proposed approach with the existing state of the art approaches. It can be observed that the proposed approach performs much better than the recently proposed algorithms in all the test sequences.



Grove2 RubberWhale

Fig. 3: Comparison of flow visualization result for Middlebury training sequence using the proposed algorithm ($right$) and ground truth ($left$)

	Army		Mequon		Schefflera		Wooden		Grove		Urban		Yosemite		Teddy	
Methods	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE	AAE	AEE
Proposed method	3.12	0.08	3.25	0.24	3.01	0.24	2.56	0.14	2.75	0.64	3.65	1.99	2.62	0.15	2.66	0.58
Deepflow flow [15]	4.49	0.12	4.26	0.28	5.96	0.44	4.89	0.26	2.98	0.81	3.26	0.38	2.09	0.11	5.83	0.93
CRT flow [13]	4.18	0.11	3.22	0.24	6.20	0.50	4.21	0.23	3.32	0.86	7.43	0.60	2.55	0.12	4.60	0.79
SIOF flow [12]	4.23	0.11	3.97	0.27	7.81	0.60	4.82	0.25	3.54	0.97	4.31	0.43	2.36	0.13	3.46	0.76
LDOF flow [4]	4.60	0.12	4.67	0.32	5.63	0.43	5.80	0.45	3.52	1.01	4.84	1.10	2.46	0.12	4.85	0.94
Classic_NL [11]	3.20	0.08	3.02	0.22	3.46	0.29	2.78	0.15	2.83	0.64	3.40	0.52	2.87	0.16	1.67	0.49

Table 3: Error comparison of, our flow with different Optical flow methods in middlebury test dataset

Fig. 2: Comparison of flow visualization results for Middlebury test dataset using the proposed algorithm ($left$) and Deepflow [15] ($right$).

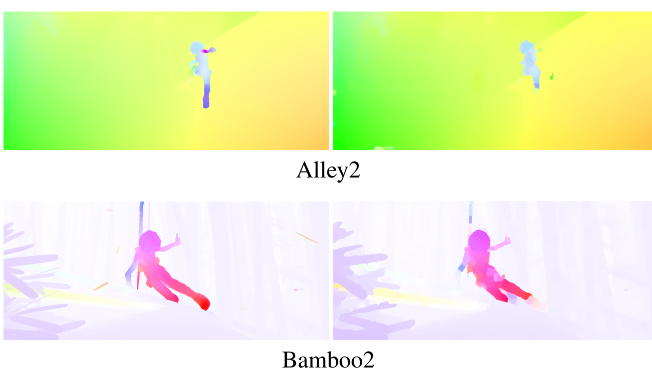
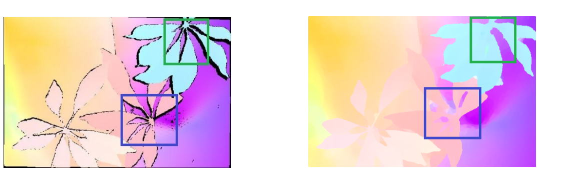


Fig. 4: Comparison of flow visualization result for MPI Sintel training sequence using the proposed algorithm ($right$) and ground truth ($left$).



(b) Groundtruth flow (b) Optical flow result obtained

Fig. 5: Comparison of Ground truth optical flow and the result obtained using the proposed. The box shows the occluded region in which the algorithm fails to capture the flow.

4.2. Qualitative Evaluation

Figure (3) shows the flow result for Middlebury training sequence using the proposed algorithm in comparison with the Ground truth. The figure is obtained using the flow color coding provided by Middlebury dataset. We have tried the algorithm on MPI Sintel dataset [24] as well. MPI Sintel datasets are used for the evaluation of large displacement optical flows. Figure (4) shows the flow result obtained using the proposed algorithm and the ground truth in MPI Sintel dataset. This shows that the algorithm is able to capture large motions as well. Comparison of Flow visualization results for the proposed algorithm with Deepflow [15] is shown in the Fig. (2). As shown in Fig. (5), the proposed algorithm has difficulties in handling corners with multiple edges. The regions shown in green and blue boxes represent areas where high error is observed.

5. CONCLUSION

In this paper we have proposed an algorithm for computing optical flow by adapting ANNF. This approach uses FM algorithm for computing the ANNF. Noisy ANNF is filtered using flow fusion. Pixel level flows obtained by flow fusion are refined using pyramidal cost optimization. The algorithm is evaluated using Middlebury test dataset. We also tried our algorithm on MPI Sintel, a large displacement optical flow dataset. The proposed algorithm shows better performance for both small and large displacement optical flow compared to the state-of-the-art algorithms.

6. REFERENCES

[1] Berthold K. P. Horn and Brian G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[2] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[3] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the international joint conference on Artificial intelligence*, 1981, vol. 2, pp. 674–679.

[4] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 2009, pp. 41–48.

[5] Jiangbo Lu, Hongsheng Yang, Dongbo Min, and Minh N. Do, "Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1854–1861.

[6] S. Avinash Ramakanth and R. Venkatesh Babu, "Synthetic image super resolution using featurematch," *Multimedia Tools and Applications*, 2014.

[7] S. Avinash Ramakanth and R. Venkatesh Babu, "Approximate nearest neighbour field based optic disk detection," *Computerized Medical Imaging and Graphics*, vol. 38, no. 1, pp. 49–56, 2014.

[8] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 24:1–24:11, 2009.

[9] S. Avinash Ramakanth and R. Venkatesh Babu, "Seamseg: Video object segmentation using patch seams," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[10] Michael 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, 1996.

[11] Deqing Sun, S. Roth, and M.J. Black, "Secrets of optical flow estimation and their principles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2432–2439.

[12] Li Xu, Zhenlong Dai, and Jiaya Jia, "Scale invariant optical flow," in *Proceedings of the European Conference on Computer Vision*, 2012, pp. 385–399.

[13] O. Demetz, D. Hafner, and J. Weickert, "The complete rank transform: A tool for accurate and morphologically invariant matching of structure," in *Proceedings of the British Machine Vision Conference*, 2013.

[14] Zhuoyuan Chen, Hailin Jin, Zhe Lin, Scott Cohen, and Ying Wu, "Large displacement optical flow from nearest neighbor fields," in *IEEE Conference on Computer Vision and Pattern Recognition.*, 2013, pp. 2443–2450.

[15] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proceedings of the Intenational Conference on Computer Vision*, 2013.

[16] Simon Korman and Shai Avidan, "Coherency sensitive hashing," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 1607–1614.

[17] S. Avinash Ramakanth and R. Venkatesh Babu, "Feature-Match: An efficient low dimensional patchmatch technique.," in *Proceedings of the Indian Conference on Computer Vision Graphics and Image Processing*, 2012, pp. 45:1–45:7.

[18] S. Avinash Ramakanth and R. Venkatesh Babu, "Feature-match: A general annf estimation technique and its applications," *IEEE Transactions on Image Processing*, 2014.

[19] G. Ben-Artzi, H. Hel-Or, and Y. Hel-Or, "The gray-code filter kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 382–393, 2007.

[20] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr, "Lucas/kanade meets horn/schunck: combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[21] Deqing Sun, Stefan Roth, and Michael J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," Tech. Rep. CS-10-03, Jan. 2013.

[22] Yingying Li and Stanley Osher, " A new median formula with applications to PDE based denoising," *Communications in Mathematical Sciences*, vol. 7(3):741-753, 2009.

[23] "Middlebury optical flow dataset," `http://vision.middlebury.edu/flow/`.

[24] "MPI Sintel Dataset," `http://sintel.is.tue.mpg.de/`.