HAL open science

Dense motion estimation between distant frames: combinatorial multi-step integration and statistical selection

Pierre-Henri Conze, Tomas Crivelli, Philippe Robert, Luce Morin

To cite this version:

Pierre-Henri Conze, Tomas Crivelli, Philippe Robert, Luce Morin. Dense motion estimation between distant frames: combinatorial multi-step integration and statistical selection. IEEE International Conference on Image Processing, Sep 2013, Melbourne, Australia. pp.3429. hal-00868251

HAL Id: hal-00868251

https://hal.science/hal-00868251v1

Submitted on 1 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DENSE MOTION ESTIMATION BETWEEN DISTANT FRAMES: COMBINATORIAL MULTI-STEP INTEGRATION AND STATISTICAL SELECTION

Pierre-Henri Conze[⋆†] *Tomás Crivelli*[⋆] *Philippe Robert*[⋆] *Luce Morin*[†]

[⋆]Technicolor
[†]INSA Rennes, IETR/UMR 6164, UEB

ABSTRACT

Accurate estimation of dense point correspondences between two distant frames of a video sequence is a challenging task. To address this problem, we present a combinatorial multi-step integration procedure which allows one to obtain a large set of candidate motion fields between the two distant frames by considering multiple motion paths across the video sequence. Given this large candidate set, we propose to perform the optimal motion vector selection by combining a global optimization stage with a new statistical processing. Instead of considering a selection only based on intrinsic motion field quality and spatial regularization, the statistical processing exploits the spatial distribution of candidates and introduces an intra-candidate quality based on forward-backward consistency. Experiments evaluate the effectiveness of our method for distant motion estimation in the context of video editing.

***Index Terms*—** motion estimation, statistical analysis, dense point matching, distant frames

1. INTRODUCTION

Despite rapid and significant progress since early formulations [1, 2], optical flow estimation still remains an open issue with crucial implications in computer vision. State-of-the-art methods [3, 4, 5, 6, 7, 8, 9] have shown to be highly accurate for estimating dense motion fields between consecutive frames of a video sequence. However, they show limitations when applied to distant frames. The classical optical flow assumptions are not verified in this case, especially for difficult situations such as illumination changes, large motion, temporal occlusions, zooming, non-rigid deformations, low color contrast and transparency.

Direct matching between distant frames can be thus sensitive to ambiguous correspondences. An alternative consists in computing the long-range displacement through concatenation of elementary optical flow fields. This can be done by temporal integration, similarly to dense point tracking algorithms [7]. However, even small errors in the input fields can lead to large drifts in the final motion field.

A first step towards accurate long-range dense correspondences is to combine numerous estimations from direct matching and temporal integration. Following a similar approach to that presented in [10], one can select for each pixel the optimal motion vector among a set of candidate motion fields based on intrinsic motion field quality and spatial regularization. A more sophisticated processing, described in [11, 12], consists in sequentially merging a set of concatenated multi-step motion fields at intermediate frames up to the target frame. However, in either case, the optimal motion vector selection strongly depends on the same optical flow assumptions that frequently fail between distant frames. This issue could be partially compensated by complexifying the matching criteria *ad-infinitum*, but an uncertainty component is always present. This argues in favor of a statistical processing which takes into account the random nature of these perturbations among a large set of dense motion fields.

In this direction, we propose two main contributions to address the dense matching problem between distant frames. Firstly, we present a combinatorial multi-step integration method which allows one to get a large set of motion fields between two distant frames by considering multiple motion *paths* across the sequence. Secondly, once this motion candidate construction is performed, we apply a new approach to select the optimal motion field based on statistics and spatial regularization. Results for motion estimation between distant frames in the context of video editing are presented.

2. MOTION CANDIDATE CONSTRUCTION

Let us consider a sequence of $N+1$ RGB images $\{I_n\}_{n\in[\![0,\dots,N]\!]}$ and let I_a and I_b be two distant frames of this sequence with $0 \leq a < b \leq N$. In this paper, we focus on the frame pair $\{I_a, I_b\}$ and our goal is to accurately estimate a dense motion field between these two frames. In this section, we aim at generating multiple motion maps between I_a and I_b.

2.1. Direct matching with multiple optical flow estimators

A first approach for building multiple motion candidates consists in considering a direct motion estimation using different optical flow methods [10]. Even if the considered estimators may fail in some regions, the idea is to pool the strengths of each one. Furthermore, the same estimator can be used several times by modifying its parameter settings. In addition,

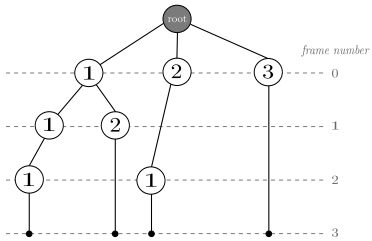


Fig. 1: Generation of *step sequences*. Going from the root node to leaf nodes of this tree structure gives $\mathbf{\Gamma}_{a,b}$, the set of possible *step sequences* from I_a to I_b.

we can derive from each version a parametric motion field for which motion is constrained by a global transformation [13]. Direct matching is more ambiguous as the distance between I_a and I_b increases. Due to a large motion range, the motion of periodic color patterns or uniform areas may not be correctly estimated. This supports a motion field construction stage using concatenation of various optical flow fields.

2.2. Combinatorial multi-step integration

Let us describe the concept of motion *path* as an alternative to direct matching for obtaining a displacement map between I_a and I_b. A motion *path* is obtained through concatenation of elementary optical flow fields across the video sequence. It links each pixel $\boldsymbol{x}_a$ of I_a to a corresponding position in I_b. Elementary optical flow fields can be computed between consecutive frames or with different frame *steps* [11, 12], i.e. with larger inter-frame distances. Let $S_n = \{s_1, s_2, \ldots, s_{Q_n}\} \subset \{1, \ldots, N-n\}$ be the set of Q_n possible *steps* at instant n. This means that the set of optical flow fields $\{\boldsymbol{v}_{n,n+s_1}, \boldsymbol{v}_{n,n+s_2}, \ldots, \boldsymbol{v}_{n,n+s_{Q_n}}\}$ is available.

Our objective is to produce a large set of motion maps between I_a and I_b as to form a significative set of samples upon which a statistical processing would be meaningful and advantageous. Given this objective, we propose to initially generate all the possible *step sequences* (i.e. combinations of *steps*) in order to join I_b from I_a. Each *step sequence* defines a motion *path*. Let $\mathbf{\Gamma}_{a,b} = \{\boldsymbol{\gamma}_0, \boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{K-1}\}$ be the set of K possible *step sequences* $\boldsymbol{\gamma}_i$ between I_a and I_b. $\mathbf{\Gamma}_{a,b}$ is computed by building a tree structure (Fig. 1) where each node corresponds to a motion field assigned to a given frame for a given *step* value (node value). In pratice, the construction of the tree is done recursively starting from I_a: we create for each node as many children as the number of *steps* available at the current instant. A child node is not generated when I_b has already been reached (therefore, the current node is considered as a leaf) or if I_b is passed given the considered *step*. Finally, once the tree has been built, going from the root node to leaf nodes gives $\mathbf{\Gamma}_{a,b}$, the set of *step sequences*. For illustration, the tree in Fig.1 indicates the four *step sequences* that can be generated going from I_0 to I_3 with *steps* 1, 2 and 3: $\mathbf{\Gamma}_{0,3} = \{\{1,1,1\},\{1,2\},\{2,1\},\{3\}\}$.

Once all the possible *step sequences* $\boldsymbol{\gamma}_i$ $\forall i \in [\![0, \ldots, K-1]\!]$ between I_a and I_b have been generated, the corresponding

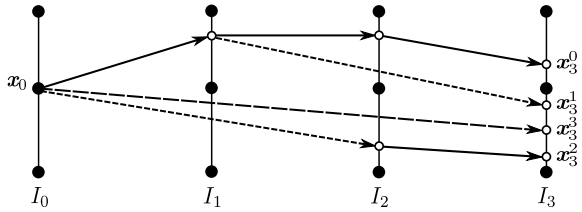


Fig. 2: Generation of multi-step motion *paths*. For each pixel $\boldsymbol{x}_0$ of I_0, this gives a set of candidate positions in I_3.

motion *paths* can be constructed through 1st-order *Euler* integration. Starting from each pixel $\boldsymbol{x}_a$ of I_a and for each *step sequence*, this integration performs the accumulation of optical flow fields following the *steps* which form the current *step sequence*. Thus, with *steps* 1, 2 and 3, Fig.2 illustrates the construction of the four possible motion *paths* (one for each *step sequence* of $\mathbf{\Gamma}_{0,3}$) between I_0 and I_3. Let $f_j^i = a + \sum_{k=0}^{j} s_k^i$ be the current frame number during the construction of motion *path* i from I_a where j is the *step* index within the *step sequence* $\boldsymbol{\gamma}_i$. For each $\boldsymbol{\gamma}_i \in \mathbf{\Gamma}_{a,b}$ and for each *step* $s_j^i \in \boldsymbol{\gamma}_i$, we start from x_a in order to iteratively compute:

$$\boldsymbol{x}_{f_j^i} = \boldsymbol{x}_{f_{j-1}^i} + \boldsymbol{v}_{f_{j-1}^i, f_j^i}(\boldsymbol{x}_{f_{j-1}^i}) \tag{1}$$

Once all the *steps* $s_j^i \in \boldsymbol{\gamma}_i$ have been run through, we obtain $\boldsymbol{x}_b^i$, the corresponding position in I_b of $\boldsymbol{x}_a$ of I_a obtained with *step sequence* $\boldsymbol{\gamma}_i$. By considering all the *step sequences*, we finally get a large set of candidate positions in I_b and this for each pixel $\boldsymbol{x}_a$ of I_a. Note that the occlusion maps attached to input motion fields are used to possibly stop the motion *path* construction. Considering an intermediate point $\boldsymbol{x}_{f_j^i}$ during the construction, a *step* can be added only if the closest pixel to $\boldsymbol{x}_{f_j^i}$ is considered as un-occluded for this *step*. Otherwise, the motion *path* is removed. In the following, the large set of candidate positions in I_b is defined as $T_{a,b}(\boldsymbol{x}_a) = \{\boldsymbol{x}_b^i\}$ $\forall i \in [\![0, \ldots, K_{\boldsymbol{x}_a} - 1]\!]$ where $K_{\boldsymbol{x}_a}$ is the cardinal of $T_{a,b}(\boldsymbol{x}_a)$.

Up to now, we have considered an exhaustive generation of *step sequences* for clarity. However, for very distant frames and for a large set of *steps*, it is not possible to consider all possible *step sequences* (computational and memory issues). For instance, for a distance of 30 frames and with *steps* 1, 2, 5 and 10, the number of possible motion *paths* is 5877241. Therefore, the procedure described above is performed on a reasonable number of *step sequences* and not for all as previouly assumed. Firstly, we limit the number of elementary vectors composing the motion *paths* by providing a maximum number of concatenations N_c. Indeed, the concatenation of numerous vectors may lead to an important drift. Secondly, we randomly select N_s motion *paths* among the remaining motion *paths* (N_s determined by storage capacity). This selection is guided by the fact that the candidate vectors should not be highly correlated. The frequency of appearance of a given *step* at a given frame must be uniform among all the possible *steps* arising from this frame in order to avoid a systematic bias towards the more populated branches of the tree.

(a) I_{25} (b) I_{40} (c) I_{55} (d) I_{46} (e) *SP* (f) *GO* (g) *SP+GO*

Fig. 3: Source frames of the *MPI S1* sequence [14] and reconstruction of the kiosk of I_{46} from I_{25} with: 1) the statistical processing (*SP*), 2) the global optimization (*GO*) method solved by *fusion moves* [15], 3) both combined (*SP+GO*).

2.3. Joint *forward* and *backward* processing

Motion estimation can be enhanced by considering both *forward* and *backward* motion fields. Similarly to the *forward* direction, the set of *backward* motion fields from each pixel $\boldsymbol{x}_b$ of I_b to I_a can be computed by considering multi-step *backward* motion *paths*. These *backward* motion fields can be inversed into *forward* motion fields in order to enrich $T_{a,b}(\boldsymbol{x}_a)$. Thus, *backward* motion vectors from pixels of I_b are projected into I_a. For each one, we identify the nearest pixel of the arrival position. The corresponding vector from I_b to I_a is reversed and started from the previously identified nearest pixel which gives a new candidate for $T_{a,b}(\boldsymbol{x}_a)$. Candidates of $T_{a,b}(\boldsymbol{x}_a)$ which have been obtained through this procedure are defined as *reverse*. Otherwise, we call them *direct*.

3. MOTION VECTOR SELECTION ON LARGE SETS

3.1. Statistical processing for motion vector selection

Given $T_{a,b}(\boldsymbol{x}_a) = \{\boldsymbol{x}_b^i\}_{i\in[\![0,\ldots,K_{\boldsymbol{x}_a}-1]\!]}$, the set of candidate positions in I_b obtained for each pixel $\boldsymbol{x}_a$ of I_a from the motion field construction stage described above, the objective is to select the optimal candidate position $\boldsymbol{x}_b^* \in T_{a,b}(\boldsymbol{x}_a)$ by exploiting the statistical information on the point distribution and the quality of each candidate. The idea is to assume a *Gaussian* model for the distribution of $T_{a,b}(\boldsymbol{x}_a)$ and try to find its central value, $\boldsymbol{x}_b^*$. Using the *maximum likelihood* estimator (MLE) and imposing the selection among elements of $T_{a,b}(\boldsymbol{x}_a)$, the choice of the optimal candidate position $\boldsymbol{x}_b^*$ is defined by:

$$\boldsymbol{x}_b^* = arg\min_{\boldsymbol{x}_b^i} \sum_{\substack{j=0\\ j\neq i}}^{K_{\boldsymbol{x}_a}-1} \left\|\boldsymbol{x}_b^j - \boldsymbol{x}_b^i\right\|_2^2 \tag{2}$$

The assumption of *Gaussianity* can be largely perturbed by outliers. Consequently, a robust estimation of the distribution central value is necessary:

$$\boldsymbol{x}_b^* = arg\min_{\boldsymbol{x}_b^i} med_{j\neq i}\left\|\boldsymbol{x}_b^j - \boldsymbol{x}_b^i\right\|_2^2 \tag{3}$$

Finally, each candidate position $\boldsymbol{x}_b^i$ receives a corresponding quality score $Q(\boldsymbol{x}_b^i)$ computed using the inconsistency values $Inc(\boldsymbol{x}_b^i)$. $Inc(\boldsymbol{x}_b^i)$ corresponds to the *Euclidean* distance to the nearest *reverse* (resp. *direct*) candidate among the distribution if $\boldsymbol{x}_b^i$ is *direct* (resp. *reverse*). We aim at assigning high quality to candidates for which the corresponding motion field between I_a and I_b is consistent with a motion field between I_b and I_a. Quality scores $Q(\boldsymbol{x}_b^i)$ are computed as follows: the maximum and minimum values of $Inc(\boldsymbol{x}_b^i)$ among all candidates are mapped from 0 to a predefined integer Q_{max}. Intermediate inconsistency values are mapped to the line defined by these two values and the result is rounded to the nearest integer: $Q(\boldsymbol{x}_b^i) \in [\![0,\ldots,Q_{max}]\!]$. The higher $Q(\boldsymbol{x}_b^i)$, the smaller $Inc(\boldsymbol{x}_b^i)$. We aim at promoting candidates in the neighborhood of high quality candidates. In practice, $Q(\boldsymbol{x}_b^i)$ is used as a voting mechanism [16]: while computing the medians in Equation (3), each sample $\boldsymbol{x}_b^j$ is considered $Q(\boldsymbol{x}_b^j)$ times to set the occurrence of elements $\left\|\boldsymbol{x}_b^j - \boldsymbol{x}_b^i\right\|_2^2$ which enforces the *forward-backward* motion consistency.

The statistical processing being applied for each pixel independently, we describe in what follows a global optimization method which includes regularization.

Frame pairs	{25,45}	{25,46}	{25,47}	{25,48}	{25,49}	{25,50}
SP	12.72	15.27	21.7	25.33	**24.48**	24.7
GO	11.19	14	11.14	13.7	21.7	22.22
SP+GO	**12.84**	**16.11**	**24.75**	**25.55**	24	**24.79**

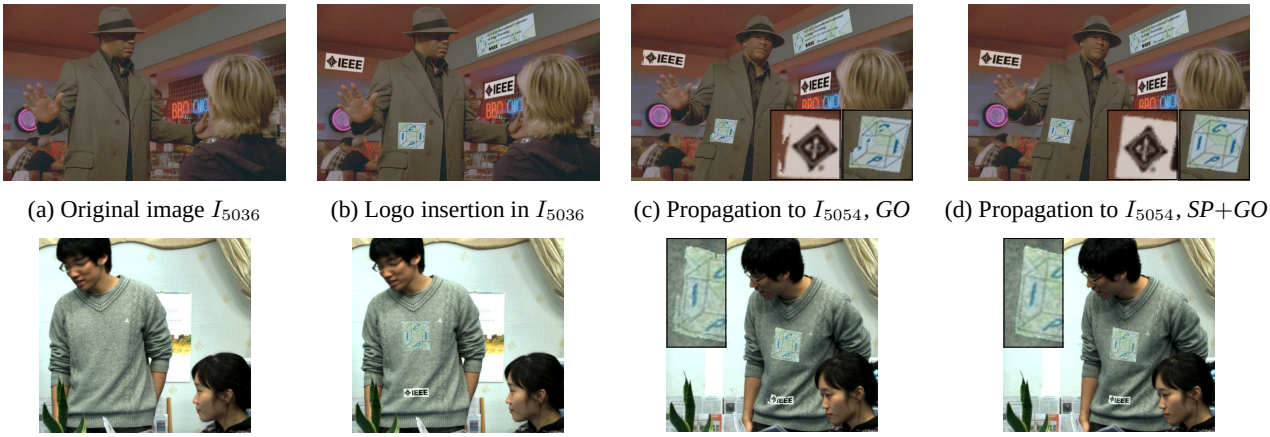
Table 1: Comparison through registration and PSNR assessment between: 1) the statistical processing (*SP*), 2) the global optimization (*GO*), 3) *SP+GO*. PSNR scores are computed on the kiosk of *MPI S1* (Fig. 3). Low PSNR for first pairs are due to the foreground object which degrades the estimation.

3.2. Global optimization for motion vector selection

We perform a global optimization stage that fuses for each pixel motion candidates into a single optimal motion field, following the approach of [10]. We introduce $L = \{l_{\boldsymbol{x}_a}\}$ as a labeling of pixels $\boldsymbol{x}_a$ of I_a where each label indicates one of the candidates of $T_{a,b}(\boldsymbol{x}_a)$. Let $d_{a,b}^{l_{\boldsymbol{x}_a}}$ be the corresponding motion vectors of candidates of $T_{a,b}(\boldsymbol{x}_a)$. We define the following energy and minimize it with *fusion moves* [10, 15].

$$E_{a,b}(L) = \sum_{\boldsymbol{x}_a} \rho_d(C(\boldsymbol{x}_a, d_{a,b}^{l_{\boldsymbol{x}_a}}(\boldsymbol{x}_a)) + Inc(\boldsymbol{x}_a + d_{a,b}^{l_{\boldsymbol{x}_a}}(\boldsymbol{x}_a))) + \sum_{<\boldsymbol{x}_a,\boldsymbol{y}_a>} \alpha_{\boldsymbol{x}_a,\boldsymbol{y}_a}.\rho_r(\left\|d_{a,b}^{l_{\boldsymbol{x}_a}}(\boldsymbol{x}_a) - d_{a,b}^{l_{\boldsymbol{y}_a}}(\boldsymbol{y}_a)\right\|_1) \tag{4}$$

The data term involves the matching cost $C(\boldsymbol{x}_a, d_{a,b}^{l_{\boldsymbol{x}_a}})$ and the inconsistency value $Inc(\boldsymbol{x}_a + d_{a,b}^{l_{\boldsymbol{x}_a}})$ which is introduced to make it more robust. The regularization term involves motion similarities with neighboring positions. α_{x_a,y_a} accounts for local color similarities in frame I_a. Functions ρ_d and ρ_r



(a) Original image I_{5036} (b) Logo insertion in I_{5036} (c) Propagation to I_{5054}, *GO* (d) Propagation to I_{5054}, *SP+GO*

(e) Original image I_{230} (f) Logo insertion in I_{230} (g) Propagation to I_{160}, *GO* (h) Propagation to I_{160}, *SP+GO*

Fig. 4: a-d) Logo insertion in I_{5036} and propagation to I_{5054} (*Hope*). e-h) Logo insertion in I_{230} and propagation to I_{160} (*Newspaper*). We compare the global optimization (*GO*) method with the statistical processing (*SP*) combined to *GO* (*SP+GO*).

Frame pairs	{160,190}	{160,200}	{160,210}	{160,220}	{160,230}
GO	21.11	19.33	18.11	17.06	16.29
SP+GO	**21.42**	**19.53**	**18.3**	**17.74**	**17.09**
MSF [12]	20.5	18.22	17.8	16.95	16.6

Table 2: Registration and PSNR assessment with: the combinatorial integration followed by the global optimization (*GO*); by the statistical processing combined to *GO* (*SP+GO*); the multi-step fusion (*MSF*) method [12]. PSNR scores are computed on the whole images of *Newspaper* (Fig. 4).

are described in [10]. *Fusion moves* algorithm fuses candidates pair by pair up to getting an optimal field $d^*_{a,b}$ but its application to a large set is limited by the computational load.

3.3. Motion vector selection framework

We propose to combine statistical processing and the above global optimization stage to combine simultaneously information about the point distribution, a robust selection based on the intrinsic motion field quality and a spatial regularization. For each $\boldsymbol{x}_a \in I_a$, the statistical processing is applied to the whole set $T_{a,b}(\boldsymbol{x}_a)$. Then, we select the N_{opt} best candidates of the distribution with the criterion of median minimization of (3). Finally, *fusion moves* algorithm fuses by pairs these N_{opt} candidates up to obtaining the best one.

4. RESULTS

Our experiments focus on frame pairs taken from three sequences: *MPI S1* [14], *Hope* and *Newspaper*. For the selected pairs, the combinatorial multi-step integration has been performed taking input elementary flow fields estimated with a 2D version of the disparity estimator of [17]. For all the experiments, the parameters are : $N_c = 7$, $N_s = 100$, $Q_{max} = 2$, $N_{opt} = 3$. Steps 1, 2, 3, 4, 5, 15 and 30 have been used.

After this motion field construction stage, we have compared three selection procedures: 1) the statistical processing (*SP*), 2) the global optimization (*GO*) method solved by *fusion moves* [15], 3) the statistical processing combined with the global optimization (*SP+GO*). Firstly, the final fields of each method have been compared through registration and PSNR assessment. For a given pair $\{I_a, I_b\}$, the final fields are used to reconstruct I_a from I_b through motion compensation and color PSNR scores are computed between I_a and the registred frame for non-occluded pixels. Tables 1 and 2 show quantitative comparisons through PSNR computed for various distances between I_a and I_b respectively on the kiosk of *MPI S1* and on whole images of *Newspaper*. An example of registration of the kiosk for a distance of 21 frames is provided Fig. 3. Results show that *SP* is better than *GO* for all pairs. The low diversity of candidates at the output of *SP* limits the effect of regularization and explains the slight improvement between *SP* and *SP+GO*. The example of Fig. 3 is interesting due to the temporary occlusion of the kiosk which is jumped by multi-step motion *paths*. For this complex situation, *SP+GO* is more adapted than *GO*. Secondly, in the context of video editing, we evaluate the accuracy of *SP+GO* and *GO* by motion compensating in I_b logos manually inserted in I_a. Fig. 4 presents results for *Hope* and *Newspaper* with a distance of 18 and 70 frames respectively. For both cases, *SP+GO* shows a clear improvement compared to *GO*.

The proposed combinatorial integration combined to *SP+GO* gives better performance compared to the multi-step fusion (*MSF*) method [12] according to PSNR scores of Table 2. The *MSF* method itself has been shown in [12] to outperform state-of-the-art methods such as [3, 6, 7].

5. CONCLUSION

We perform long-term dense matching by considering multiple multi-step motion *paths* along the sequence. Given the resulting large set of motion candidates, we apply a selection procedure where the global optimization stage is preceded by a new statistical processing which exploits the spatial distribution and the intrinsic quality of candidates. It leads to better results compared to state-of-the-art methods.

6. REFERENCES

[1] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *International joint conference on artificial intelligence*, vol. 2, pp. 674–679, 1981.

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

[3] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," *Pattern Recognition*, pp. 214–223, 2007.

[4] Frank Steinbrucker, Thomas Pock, and Daniel Cremers, "Large displacement optical flow computation without warping," in *IEEE International Conference on Computer Vision*, 2009, pp. 1609–1614.

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

[6] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," *European Conference on Computer Vision*, pp. 438–451, 2010.

[7] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[8] Li Xu, Jiaya Jia, and Yasuyuki Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, 2012.

[9] M. W. Tao, J. Bai, P. Kohli, and S. Paris, "SimpleFlow: A non-iterative, sublinear optical flow algorithm," *Computer Graphics Forum (Eurographics 2012)*, vol. 31, no. 2, 2012.

[10] V. Lempitsky, S. Roth, and C. Rother, "FusionFlow: Discrete-continuous optimization for optical flow estimation," *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[11] T. Crivelli, P.-H. Conze, P. Robert, and P. Pérez, "From optical flow to dense long term correspondences," in *IEEE International Conference on Image Processing*, 2012.

[12] T. Crivelli, P.-H. Conze, P. Robert, M. Fradet, and P. Pérez, "Multi-step flow fusion: Towards accurate and dense correspondences in long video shots," in *British Machine Vision Conference*, 2012.

[13] J. Wills and S. Belongie, "A feature-based approach for determining dense long range correspondences," *European Conference on Computer Vision*, pp. 170–182, 2004.

[14] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt, "MPI-S1," `http://www.mpi-inf.mpg.de/~granados/projects/vidbginp/index.html`.

[15] V. Lempitsky, C. Rother, S. Roth, and A. Blake, "Fusion moves for Markov random field optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1392–1405, 2010.

[16] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: a tutorial," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 43, no. 3, pp. 157–192, 1996.

[17] P. Robert, C. Thébault, V. Drazic, and P.-H. Conze, "Disparity-compensated view synthesis for s3d content correction," in *SPIE IS&T Electronic Imaging Stereoscopic Displays and Applications*, 2012.