MOTION ESTIMATION VIA HIERARCHICAL BLOCK MATCHING AND GRAPH CUT

Amin Zheng, Yuan Yuan, Sunil Prasad Jaiswal and Oscar C. Au

The Hong Kong University of Science and Technology
Emails: {amzheng, yyuanad, spjaiswal, eeau}@ust.hk

ABSTRACT

Block matching based motion estimation algorithm is adopted in numerous practical video processing applications due to its low complexity. However, conventional block matching based methods process each block independently to minimize the energy function, which results in a local minimum. It fails to preserve the motion details. In this paper, we formulate the motion estimation as a labeling problem. The candidate labels are initialized by adopting a hierarchical block matching method. Then, we employ a graph cut algorithm to efficiently solve the global labeling problem with candidate labels. Experimental results show that the proposed approach can well preserve the motion details and outperforms all other block based motion estimation methods in term of endpoint error and angle error on the Middleburry optical flow benchmark.

Index Terms— motion estimation, block matching, graph cut, optical flow

1. INTRODUCTION

Accurate motion estimation is essential for a variety of applications such as frame-rate conversion, object tracking, video compression and super resolution. The estimated motion fields are able to describe the movement trend of the objects in current frame, which allows for the prediction of future frames. However, finding the true motion fields is non-trivial because of the aperture problem, occlusion, illumination change, and so on [1]. To alleviate the impact of these problems, motion estimation methods are typically modeled by a set of energy functions which contain different kinds of regularization terms along with a brightness comparison term.

The brightness comparison is determined by measuring the pixel or block correlation [2]. Most regularization terms impose the smoothness constraints which are based on the assumption that the motion fields are piece-wise smooth [1]. A proper regularization term is effective to avoid the aperture and occlusion problems. As the smoothness constraints enforce a connection between neighboring regions, the energy function minimization is thus a global optimization problem which can be solved either by continuous optimization or discrete optimization algorithms. The continuous optimization algorithms are used in most optical flow based motion estimation methods [3] that the energy is minimized by solving the Euler-Lagrange differential equations. These methods are able to provide high quality motion fields, but they need sophisticated energy terms to accurately model the smoothness and the equations are solved at the expense of high computational complexity.

The energy minimization can also be regarded as a discrete optimization problem that each node (pixel or block) need to be assigned a label (motion vector) in order to minimize the total energy. Some well-known discrete optimization algorithms, such as graph cut [4] and belief propagation [5], can well handle this labeling problem. Unfortunately, the original $2D$ label space is so large that the computation complexity can not be afforded in most applications. Block matching methods [6–8] are also one kind of discrete optimization which have relatively low complexity. Each block is assigned a motion vector by searching the matching block in the reference frame to minimize the sum of absolute difference (SAD). In [6], the motion fields are estimated in a recursive manner without considering the smoothness constraint. In [7] and [8], smoothness terms are incorporated into the energy function along with the SAD term for performing block matching. However, in both [7] and [8], although they impose the smoothness constraints, each node is processed one by one to minimize its own energy function independently. Thus, the obtained solution is a local minimum.

In this paper, we propose a motion estimation approach based on block matching and graph cut. Firstly, we model the motion estimation process as a labeling discrete optimization problem, which minimize the cost function containing a brightness constancy term and a weighted smoothness regularization term. Secondly, we use block matching to initialize candidate labels of each node, then the labeling problem with candidate labels is solved efficiently in a hierarchical manner by the graph cut algorithm. Thirdly, we adopt a occlusion handling scheme to deal with occluded pixels. Experiments show that the proposed method outperforms multiple optical flow methods with much higher complexity on the Middlebury benchmark [3] website. In terms of the endpoint error and angle error, our approach outperforms all the other block based methods.

The rest of the paper is organized as follows. We formulate the motion estimation as a labeling problem in Section

2. In section 3, we show how to solve this labeling problem via hierarchical block matching and graph cut. Experimental results and further analysis are given in Section 4. Section 5 concludes this paper.

2. PROBLEM FORMULATION

The general energy function of motion estimation problem is as follows:

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

where $E_D(\boldsymbol{u})$ is the data term that measures the similarity between current frame and predicted frame given the motion fields denoted by $\boldsymbol{u}$. $E_S(\boldsymbol{u})$ is the regularization term which penalizes deviation in the smoothness of the motion fields. And λ is a regularization weight.

As the proposed method estimates the motion in a hierarchical manner with reduced block size in each level, the node in the energy function can be a pixel or block. Next we will specify the proposed data term and regularization term.

2.1. Data term

Our data term is based on the brightness constancy assumption. And l_1 norm is used because of its robustness compared to l_2 norm [9]. The data term is given as follows:

$$E_D(\boldsymbol{u}) = \sum_{\mathbf{x}} \left| \frac{1}{N} \sum_{\mathbf{i} \in \mathcal{B}_{\mathbf{x}}} |\boldsymbol{I}_0(\mathbf{i} + \boldsymbol{u}(\mathbf{x})) - \boldsymbol{I}(\mathbf{i})| \right| \quad (2)$$

where $\mathbf{x} \in \mathcal{L}$ is the $2D$ block index and $\mathcal{B}_{\mathbf{x}}$ is a set containing all the pixel indexes in the block indicated by $\mathbf{x}$. N denotes the number of pixels in the block. $\boldsymbol{I}$ and $\boldsymbol{I_0}$ denote the current frame and the reference frame, respectively. $\boldsymbol{u}(\mathbf{x})$ is the motion vector of node $\mathbf{x}$. Specifically, if the block size is 1×1 (pixel), (2) is simplified as follows:

$$E_D(\boldsymbol{u}) = \sum_{\mathbf{x}} \left| \boldsymbol{I_0}(\mathrm{x} + \boldsymbol{u}(\mathbf{x})) - \boldsymbol{I}(\mathrm{x}) \right| \quad (3)$$

2.2. Regularization term

Besides the smoothness constraint, the regularization term for motion estimation problem is generally designed to be edge preserving. Thus, we define the regularization term as follows:

$$E_S(\boldsymbol{u}) = \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}} w(\mathbf{x}, \mathbf{y}) \Big|\Big| \boldsymbol{u}(\mathbf{x}) - \boldsymbol{u}(\mathbf{y}) \Big|\Big|_1 \quad (4)$$

where $\mathcal{N}$ is an eight-neighbors system on nodes. $||\boldsymbol{u}(\mathbf{x}) - \boldsymbol{u}(\mathbf{y})||_1$ measures the smoothness of motion fields. $w(\mathbf{x}, \mathbf{y})$ is a structure adaptive weight which maintains the motion discontinuity [10]:

$$w(\mathbf{x}, \mathbf{y}) = exp\Big(- \Big| \frac{1}{N} \sum_{\mathbf{i} \in \mathcal{B}_{\mathbf{x}}} \boldsymbol{I}(\mathbf{i}) - \frac{1}{N} \sum_{\mathbf{j} \in \mathcal{B}_{\mathbf{y}}} \boldsymbol{I}(\mathbf{j}) \Big|^{\kappa} \Big) \quad (5)$$

where we set $\kappa = 0.8$ in our experiments.

Thus, we get the final energy minimization problem:

$$\begin{aligned} \min_{\boldsymbol{u}} \quad & \sum_{\mathbf{x}} \Big| \tfrac{1}{N} \sum_{\mathbf{i} \in \mathcal{B}_{\mathbf{x}}} |\boldsymbol{I}_0(\mathbf{i} + \boldsymbol{u}(\mathbf{x})) - \boldsymbol{I}(\mathbf{i})| \Big| \\ & + \lambda \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}} w(\mathbf{x}, \mathbf{y}) \Big|\Big| \boldsymbol{u}(\mathbf{x}) - \boldsymbol{u}(\mathbf{y}) \Big|\Big|_1 \\ \text{s.t.} \quad & \boldsymbol{u}(\mathbf{x}) \in \mathcal{F}(\mathbf{x}) \end{aligned} \quad (6)$$

where $\mathcal{F}(\mathbf{x})$ denotes the set of feasible motion vectors of node $\mathbf{x}$. When the node is pixel, by allowing the elements of $\mathcal{F}(\mathbf{x})$ be real numbers, (6) is a typical optical flow problem [3].

In our problem, we assume $\boldsymbol{u}(\mathbf{x})$ be quarter pixel precision, then the number of feasible motion vectors is finite. Thus, (6) can be regarded as a labeling problem that each node $\mathbf{x}$ need to be labeled a motion vector from the candidate set $\mathcal{F}(\mathbf{x})$ in order to minimize the energy function. However, the possible candidates of $\boldsymbol{u}(\mathbf{x})$ for each node have $4M \times 4N$ choices, where M and N denote the frame height and width, respectively. It is time consuming to solve this labeling problem directly. Besides, each candidate set contains too many redundancy motion vectors which increases the probability of getting outliers. In next section, we employ block matching to reduce the number of elements in each candidate set to nine, then the labeling problem with candidate labels is solved by graph cut algorithms.

3. OPTIMIZATION FRAMEWORK

We estimate the motion fields in a hierarchical manner starts at the block size of 64×64 which is halved in six iterations until to the pixel level. When the block size is larger than 4×4, block matching is used to get the initialized motion vectors which equals to $\lambda = 0$ in (6). As the smoothness constraint does not always hold for large block, it is reasonable without considering the regularization term. For the smaller block, graph cut algorithm is adopted to solve the global discrete optimization problem.

3.1. Motion vector initialization via block matching

In [11], the motion is hierarchically estimated using decreased block sizes at each level. The hierarchical block matching strategy is able to prevent local minimum. We use the same idea in this paper to get the initialized motion vectors.

At each hierarchy level, the motion vectors are computed using conventional block matching algorithm by minimizing the mean of absolute difference (MAD). The search area of current level is dependent on the motion vectors of previous level, as illustrated in Fig. 1. The set of starting search points of current block is formed on the nine neighboring motion vectors of previous level, and the search range around each starting points is decreased with each level. Thus, the motion of current level is able to follow the motion of the previous level. After the search is finished at each level, the motion

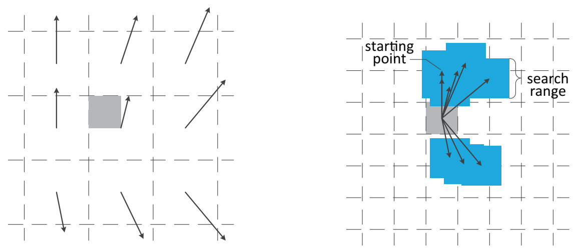


Fig. 1. Search area (blue) of current level derived from previous level.

fields are smoothed by using a vector median filter to eliminate outliers.

To reduce the effect of noise caused by small block matching, the search strategy is switched to solving the labeling problem in (6) when the block size is smaller than 8×8. The motion vector candidate set $\mathcal{F}(\mathbf{x})$ of current level is constituted by the nine neighboring motion vectors of previous level. By solving (6), we get the motion vectors of current level which define the candidate set of next level in a same pattern. This forces the small blocks to decide to which motion object they belong. The process is repeated until to the pixel level.

3.2. Energy minimization via graph cut

Graph cut algorithm [4] is adopted to solve the labeling problem. The standard form of energy function in graph cut problem is as follows:

$$E(\boldsymbol{u}) = \sum_{\mathbf{x}} D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x})) + \sum_{\mathbf{x},\mathbf{y} \in \mathcal{N}} V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y})) \quad (7)$$

where $D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x}))$ measures the cost of assigning the label $\boldsymbol{u}(\mathbf{x})$ to the node $\mathbf{x}$. $V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y}))$ measures the cost of assigning the labels $\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y})$ to the adjacent nodes $\mathbf{x}, \mathbf{y}$ and is used to impose spatial smoothness.

In our problem (6), $D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x}))$ and $V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y}))$ are separately derived as follows:

$$\begin{aligned} D_{\mathbf{x}}(\boldsymbol{u}(\mathbf{x})) &= \left| \frac{1}{N} \sum_{\mathbf{i} \in \mathcal{B}_{\mathbf{x}}} |\boldsymbol{I}_0(\mathbf{i} + \boldsymbol{u}(\mathbf{x})) - \boldsymbol{I}(\mathbf{i})| \right| \\ V_{\mathbf{x},\mathbf{y}}(\boldsymbol{u}(\mathbf{x}), \boldsymbol{u}(\mathbf{y})) &= \lambda w(\mathbf{x}, \mathbf{y}) \left|\left| \boldsymbol{u}(\mathbf{x}) - \boldsymbol{u}(\mathbf{y}) \right|\right|_1 \end{aligned} \quad (8)$$

After obtaining the candidate set $\mathcal{F}(\mathbf{x})$ at each level when the block size is smaller than 8×8, we use the graph cut algorithm solve (6) hierarchically to get the pixel level motion fields.

3.3. Occlusion handling

Occlusion is a common problem for motion estimation. The motion vectors of occluded pixels generally can not be determined due to the lack of correspondence. The brightness constancy assumption will not hold any more. In this section, we handle the occlusion based on the estimated pixel level motion fields. We first detect the occluded pixels based on the observation that, multiple pixels mapping to the same point in the target frame using forward warping are possibly occluded by each other [12]. Then, a weighted data term based on the occlusion detection is proposed which updates the energy function as follows:

$$E'(\boldsymbol{u}) = c(\mathbf{x}) E_D(\boldsymbol{u}) + \lambda E_S(\boldsymbol{u}) \quad (9)$$

where $c(\mathbf{x}) = \max(1 - o(\mathbf{x}), 0.05)$ is the weight and $o(\mathbf{x})$ is the occlusion label. The label is set, i.e., $o(\mathbf{x}) = 1$ if there exist more than one reference pixel mapping to position $\mathbf{x} + \boldsymbol{u}(\mathbf{x})$ in the target frame, otherwise $o(\mathbf{x}) = 0$. This simple occlusion detection method tends to fatten the occlusion region, but it seldom leaves out the occluded pixels.

When the pixel is occluded, we should not trust the data term but depend on the smoothness term to estimate the motion vector. $c(\mathbf{x})$ is set to at least 0.05 to make the energy function more robust to noise. The updated energy function is solved by the graph cut algorithm to get the final motion fields. The occlusion handling process is performed as the last step after obtaining the pixel level motion fields.

Table 1. Results of proposed approach compared to [8].

Sequence	Endpoint Error		Improv. [dB]
	BlockOverlap [8]	Proposed	
`Dimetrodon`	0.215	0.134	2.05
`Grove2`	0.202	0.168	0.80
`Grove3`	0.618	0.483	1.07
`Hydrangea`	0.230	0.198	0.65
`Rubber Whale`	0.161	0.134	0.80
`Urban 2`	0.418	0.278	1.77
`Urban 3`	0.662	0.697	-0.22
`Venus`	0.315	0.261	0.82
Average	0.353	0.294	0.79

4. EXPERIMENTS

In all the experiments, we apply the Rudin-Osher-Fatemi(ROF) structure texture decomposition method [13] to pre-process the input sequences to alleviate the impact of lighting change. We adopt a sub-pixel block matching scheme [6] to get the quarter pixel precision motion vectors when the block size is 8×8. Based on the experiments, λ is set to 0.05, 0.1 and 0.2 for block size 4×4, 2×2 and 1×1, respectively.

We evaluate the performance of proposed approach on the Middlebury datasets. Table 1 shows the endpoint error results of the eight ground truth test sequences comparing to [8] which is one of the best block-based motion estimation methods. As shown in Table 1, our proposed method results in an average of 0.79 dB improvement. The improvement partly comes from the more accurate regularization term and the discrete global optimization scheme in the proposed method.

We also submitted our results to the Middlebury online benchmark for comparison with other motion estimation al-

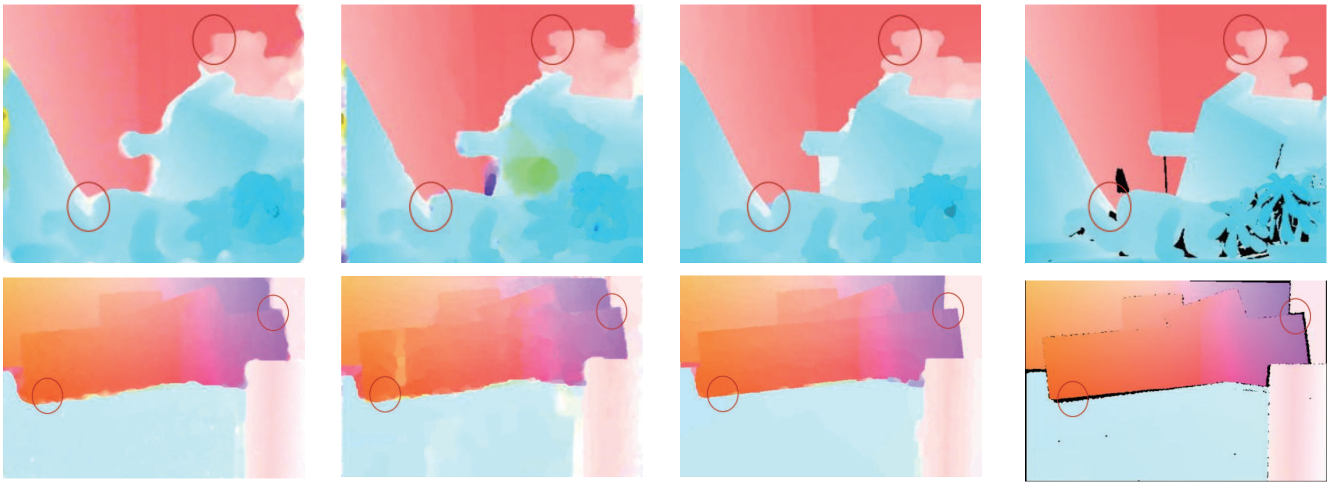
Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
ComplOF-FED-GPU [35]	49.0	0.11 48	0.29 54	0.10 66	0.21 27	0.78 50	0.14 16	0.32 36	0.79 39	0.17 13	0.19 48	0.99 56	0.11 46	0.89 55	1.29 57	0.73 54	1.25 78	1.74 75	0.64 73	0.14 42	0.13 27	0.30 71	0.64 45	1.50 54	0.83 46
Classic++ [32]	50.4	0.09 29	0.25 32	0.07 14	0.23 45	0.78 50	0.19 46	0.43 50	1.00 53	0.22 50	0.20 53	1.11 62	0.10 42	0.87 50	1.30 59	0.66 46	0.47 36	1.62 66	0.33 37	0.17 73	0.14 49	0.32 81	0.79 66	1.64 63	0.92 57
HBM-GC [106]	50.4	0.14 76	0.28 47	0.12 78	0.26 59	0.69 31	0.22 65	0.34 38	0.75 34	0.22 50	0.21 58	0.77 23	0.15 65	0.67 26	0.97 23	0.52 31	0.63 60	0.81 6	0.44 57	0.22 92	0.19 90	0.36 86	0.54 33	1.21 41	0.78 41
Aniso. Huber-L1 [22]	51.1	0.10 41	0.28 47	0.08 38	0.31 70	0.88 64	0.28 75	0.56 69	1.13 62	0.29 68	0.20 53	0.92 51	0.13 56	0.84 47	1.20 45	0.70 49	0.39 19	1.23 38	0.28 14	0.17 73	0.15 59	0.27 57	0.64 45	1.36 45	0.79 42
TF+OM [100]	52.2	0.10 41	0.26 38	0.07 14	0.22 33	0.66 25	0.19 46	0.36 42	0.78 37	0.39 76	0.20 53	0.89 47	0.13 56	0.98 71	1.31 63	1.03 75	0.56 54	1.55 63	0.33 37	0.16 66	0.17 76	0.27 57	0.76 58	1.59 61	0.98 64
CRTflow [80]	53.1	0.11 48	0.30 59	0.08 38	0.24 50	0.77 47	0.17 27	0.50 60	1.13 62	0.21 45	0.23 63	1.24 69	0.12 52	0.86 49	1.27 54	0.65 45	0.60 57	1.95 84	0.50 63	0.12 22	0.14 49	0.21 31	0.79 66	1.77 70	0.98 64
DeepFlow [86]	53.1	0.12 66	0.31 62	0.11 73	0.28 65	0.82 58	0.22 65	0.44 56	1.00 53	0.33 73	0.26 70	1.34 77	0.15 65	0.81 43	1.21 46	0.58 39	0.38 16	1.55 63	0.25 8	0.11 11	**0.11** 1	0.24 44	0.93 74	1.82 75	1.12 72

(a) Average endpoint error

Average angle error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
RFlow [90]	49.0	3.82 42	10.0 45	3.44 57	2.61 22	9.73 31	2.02 19	5.66 56	14.5 56	2.05 21	3.93 60	23.1 76	1.90 50	3.24 47	4.19 49	2.66 51	4.12 55	15.2 64	3.34 48	2.61 44	3.56 29	2.65 45	4.48 66	10.5 74	3.93 68
TCOF [69]	49.0	4.17 54	10.4 52	3.71 66	3.17 44	10.7 44	2.59 53	6.58 64	15.7 64	3.82 67	3.69 56	16.1 45	2.37 63	3.78 74	4.95 86	2.47 38	2.59 6	8.47 14	2.58 15	3.66 80	4.83 77	2.67 46	1.83 26	4.20 30	1.46 13
HBM-GC [106]	49.9	5.25 76	10.5 53	4.34 77	3.17 44	8.78 21	2.94 62	4.38 39	10.6 35	2.68 47	3.59 51	12.8 9	2.47 66	2.96 32	3.64 21	2.64 49	3.96 50	8.26 10	3.56 53	4.40 92	5.92 96	3.62 82	2.55 39	6.34 43	3.29 51
ComplOF-FED-GPU [35]	51.8	4.28 61	11.3 61	3.70 65	3.25 51	13.0 64	2.16 23	4.06 37	11.2 39	1.95 18	3.91 59	19.2 58	2.01 52	3.20 43	4.15 46	2.64 49	4.61 62	16.1 69	3.90 61	2.98 65	3.77 41	3.69 83	2.85 44	7.44 52	2.53 39
SRR-TVOF-NL [91]	51.9	4.47 66	10.9 55	3.32 51	4.04 66	13.2 67	2.90 60	4.81 44	12.5 45	3.15 57	3.33 45	15.3 33	1.61 40	3.24 47	4.03 42	2.70 53	3.94 48	11.8 39	3.33 46	4.16 88	5.21 86	3.44 80	2.06 33	3.48 20	2.42 35
TF+OM [100]	53.7	3.97 47	10.2 47	2.94 38	2.91 35	9.12 26	2.57 52	5.22 50	11.5 40	6.92 76	3.59 51	16.1 45	2.28 60	3.20 43	3.97 37	3.11 64	4.70 66	14.5 59	4.32 70	3.06 68	4.84 78	2.71 48	3.93 58	8.79 59	4.32 72
Aniso. Huber-L1 [22]	53.8	3.71 41	10.1 46	3.08 43	4.36 71	13.0 64	3.77 74	6.92 68	15.3 60	3.60 65	3.54 49	15.9 43	2.04 53	3.38 56	4.45 62	2.47 38	3.88 45	12.9 46	2.74 25	3.37 73	4.36 66	2.85 56	3.16 49	7.52 53	2.90 45

(b) Average angle error

Fig. 2. Screenshots, taken at the time of writing, of the Middlebury optical flow benchmark with our methods highlight in red. The latest result tables are available at http://vision.middlebury.edu/flow/eval/



Bartels [7] BlockOverlap [8] Proposed Ground truth

Fig. 3. Visual results of `Teddy` and `Wooden`, comparing to two block matching based methods [7] and [8]. The colored motion fields are from the Middelburry optical flow benchmark. Occlusion areas are indicated with black in the ground truth images.

gorithms. There are in total 106 methods on the website at the time of writing and our proposed method denoted by HBM-GC ranks in the middle indicated by avg. rank, as shown in Fig.2. From the website, the proposed approach denoted by HBM-GC get a better result than multiple optical flow methods in terms of end point error and angle error. For these two errors, the proposed approach outperforms all other block based methods. The visual results of sequences `Wooden` and `Teddy` are given in Fig.3. We can see that the motion details is much better preserved in the results of proposed approach. The occlusion problem is also well handled at the boundary of the wood. The computation complexity of proposed method is lower than most of the optical flow based methods, but it is higher compared to [7] and [8] because of the discrete optimization. Our future work is to exploit a more efficient discrete optimization algorithm to solve the labeling problem with initialized labels.

5. CONCLUSION

In this paper, we present a motion estimation approach based on block matching and graph cut. By formulating the motion estimation as a labeling problem, we solve it with block matching initialized candidate labels using graph cut algorithm. The desired energy function well preserves the motion details. The published results for the Middlebury datasets demonstrates that the proposed approach performs well compared to other state-of-art methods.

6. REFERENCES

[1] B. K. P. Horn and B. Schunk, "Determining optical flow," in *Artif. Intell.*, 1981, vol. 17, pp. 185–203.

[2] B. Furht and B. Furht, *Motion Estimation Algorithms for Video Compression*, Kluwer, Norwell, MA, 1996.

[3] S. Baker, S. Roth, D. Scharstein, M.J. Black, J.P. Lewis and R. Szeliski, "A Database and Evaluation Methodology for Optical Flow," in *Proc. IEEE Int. Conf. Computer Vision*, Oct 2007, pp. 1–8.

[4] Y. Boykov, O. Veksler and R. Zabih, "Fast approximate energy minimization via graph cuts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1999, vol. 1, pp. 377–384.

[5] J. Yedidia, W. Freeman and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring Artificial Intelligence in the New Millennium*, vol. 8, pp. 236–239, 2003.

[6] G. De Haan and P. W. A. C. Biezen, "Sub-pixel motion estimation with 3-D recursive search block- matching," *Signal Process.*, vol. 6, no. 3, pp. 229–239, Jun 1994.

[7] C. Bartels and G. de Haan, "Smoothness constraints in recursive search motion estimation for picture rate conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 10, pp. 1310–1319, Oct 2010.

[8] M. Santoro, G. AlRegib and Y. Altunbasak, "Joint Framework for Motion Validity and Estimation Using Block Overlap," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1610–1619, Apr 2013.

[9] D. Q. Sun, S. Roth and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2010, pp. 2432–2439.

[10] A. Wedel, D. Cremers, T. Pock and H. Bischof, "Structure-and motion-adaptive regularization for high accuracy optic flow," in *Proc. IEEE Int. Conf. Computer Vision*, 2009, pp. 1663–1668.

[11] S. Klomp, M. Munderloh and J. Ostermann, "Decoder-side hierarchical motion estimation for dense vector fields," in *Proc. IEEE Int. Picture Coding Symp.*, Dec 2010, pp. 362–365.

[12] L. Xu, J. Y. 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.

[13] A. Wedel, Y. Pock, C. Zach, H. Bischof and D. Cremers, "An improved algorithm for TV-L1 optical flow," in *Proc. Dagstuhl Visual Motion Analysis Workshop*, 2008, pp. 23–45.