Supplemental Material
-
Adaptive Integration of Feature Matches into Variational Optical Flow Methods

Michael Stoll, Sebastian Volz, and Andrés Bruhn

Institute for Visualisation and Interactive Systems
Universitätsstraße 38, University of Stuttgart, Germany
{stoll,volz,bruhn}@vis.uni-stuttgart.de

1 Minimization

In this section we give additional information on the minimization of the energy of our ALD-method.

1.1 Original Energy

Let us now deduce the Euler-Lagrange equations that have to be solved on each level of the incremental coarse-to-fine warping scheme. We begin with the original energy functional, that reads

$$\begin{aligned}\mathcal{E}(du,dv) = \int_\Omega \Bigg[&\Psi_d\left(\sum_{c=1}^{3}\left|g_2^c(\boldsymbol{x}+\boldsymbol{w}) - g_1^c(\boldsymbol{x})\right|^2\right) \\ &+ \lambda\cdot\Psi_d\left(\sum_{c=1}^{3}\left|\nabla g_2^c(\boldsymbol{x}+\boldsymbol{w}) - \nabla g_1^c(\boldsymbol{x})\right|^2\right) \\ &+ \alpha\cdot\Psi_{s_1}\left(\left(\boldsymbol{r}_1^\top\nabla u\right)^2 + \left(\boldsymbol{r}_1^\top\nabla v\right)^2\right) \\ &+ \alpha\cdot\Psi_{s_2}\left(\left(\boldsymbol{r}_2^\top\nabla u\right)^2 + \left(\boldsymbol{r}_2^\top\nabla v\right)^2\right) \\ &+ \beta\cdot\chi_p\cdot c_p\cdot\Psi_p\left((u-\bar{u})^2 + (v-\bar{v})^2\right)\Bigg]\, d\boldsymbol{x}\end{aligned} \tag{1}$$

1.2 Euler-Lagrange Equations

Obviously, the contribution of the data term is not convex with respect to the estimates u and v. Following [4], one could linearize the data term directly, leading to a convex term. However, the linearization is only valid for small displacements. Thus, the authors of [2] propose to postpone the linearization and compute the

Euler-Lagrange equations first. Follwing this idea, we obtain

$$\begin{aligned}
0 = {} & \Psi'_{d,g} \cdot \sum_{c=1}^{3} \left(g_2^c(\boldsymbol{x}+\boldsymbol{w}) - g_1^c(\boldsymbol{x})\right) \cdot \partial_x g_2^c(\boldsymbol{x}) && (2)\\
& +\lambda \cdot \Psi'_{d,\nabla g} \cdot \sum_{c=1}^{3} \left(\partial_x g_2^c(\boldsymbol{x}+\boldsymbol{w}) - \partial_x g_1^c(\boldsymbol{x})\right) \cdot \partial_{xx} g_2^c(\boldsymbol{x})\\
& +\lambda \cdot \Psi'_{d,\nabla g} \cdot \sum_{c=1}^{3} \left(\partial_y g_2^c(\boldsymbol{x}+\boldsymbol{w}) - \partial_y g_1^c(\boldsymbol{x})\right) \cdot \partial_{yx} g_2^c(\boldsymbol{x})\\
& -\alpha \cdot \operatorname{div}\left(\mathbf{D}\nabla u\right) + \beta \cdot \chi_p \cdot c_p \cdot \Psi'_p \cdot (u - \bar{u})\\
0 = {} & \Psi'_{d,g} \cdot \sum_{c=1}^{3} \left(g_2^c(\boldsymbol{x}+\boldsymbol{w}) - g_1^c(\boldsymbol{x})\right) \cdot \partial_y g_2^c(\boldsymbol{x}) && (3)\\
& +\lambda \cdot \Psi'_{d,\nabla g} \cdot \sum_{c=1}^{3} \left(\partial_x g_2^c(\boldsymbol{x}+\boldsymbol{w}) - \partial_x g_1^c(\boldsymbol{x})\right) \cdot \partial_{xy} g_2^c(\boldsymbol{x})\\
& +\lambda \cdot \Psi'_{d,\nabla g} \cdot \sum_{c=1}^{3} \left(\partial_y g_2^c(\boldsymbol{x}+\boldsymbol{w}) - \partial_y g_1^c(\boldsymbol{x})\right) \cdot \partial_{yy} g_2^c(\boldsymbol{x})\\
& -\alpha \cdot \operatorname{div}\left(\mathbf{D}\nabla v\right) + \beta \cdot \chi_p \cdot c_p \cdot \Psi'_p \cdot (v - \bar{v})
\end{aligned}$$

with

$$\Psi'_{d,f} = \Psi'_d \left(\left(f(\boldsymbol{x}+\boldsymbol{w}) - f(\boldsymbol{x})\right)^2 \right) \tag{4}$$

$$\mathbf{D} = \begin{pmatrix} \Psi'_{s1} & 0 \\ 0 & \Psi'_{s2} \end{pmatrix} \tag{5}$$

$$\Psi'_{s1} = \Psi'_{s1} \left(\left(\boldsymbol{r}_1^\top \nabla u\right)^2 + \left(\boldsymbol{r}_1^\top \nabla v\right)^2 \right) \tag{6}$$

$$\Psi'_{s2} = \Psi'_{s1} \left(\left(\boldsymbol{r}_2^\top \nabla u\right)^2 + \left(\boldsymbol{r}_2^\top \nabla v\right)^2 \right) \tag{7}$$

$$\Psi'_{p} = \Psi'_{p} \left((u - \bar{u})^2 + (v - \bar{v})^2 \right) \tag{8}$$

1.3 Incremental Multi-Scale Strategy

In analogy to [2], we refrain from linearizations in the data term and compute the estimates u, v by a fixed point iteration combined with an incremental multi-scale strategy. Splitting the unknown flow $\boldsymbol{w}^{k+1}$ on each scale k into a known part $\boldsymbol{w}^k$ from the previous scale and an unknown increment $d\boldsymbol{w}^k$ from the current scale, we only have to estimate the increment. This allows to linearize the data term with respect to the increments and thus to make use of the motion tensor notation [3]. Similarly, we can use the diffusion tensor notation from [5] for the smoothness term.

1.4 Incremental Euler-Lagrange Equations

Now, we can write the equations that have to be solved on each scale as

$$\begin{aligned}0 =& \Psi_{d,g}^{k\,\prime} \cdot \left(J_{11,g}^k du^k + J_{12,g}^k dv^k + J_{13,g}^k\right)\\ &+\lambda \cdot \Psi_{d,\nabla g}^{k\,\prime} \cdot \left(J_{11,\nabla g}^k du^k + J_{12,\nabla g}^k dv^k + J_{13,\nabla g}^k\right)\\ &-\alpha \cdot \operatorname{div}\left(\mathbf{D}\nabla u\right) + \beta \cdot \chi_p \cdot c_p \cdot \Psi_p^{k\,\prime} \cdot \left(u^k + du^k - \bar{u}\right)\end{aligned} \tag{9}$$

$$\begin{aligned}0 =& \Psi_{d,g}^{k\,\prime} \cdot \left(J_{12,g}^k du^k + J_{22,g}^k dv^k + J_{23,g}^k\right)\\ &+\lambda \cdot \Psi_{d,\nabla g}^{k\,\prime} \cdot \left(J_{12,\nabla g}^k du^k + J_{22,\nabla g}^k dv^k + J_{23,\nabla g}^k\right)\\ &-\alpha \cdot \operatorname{div}\left(\mathbf{D}\nabla v\right) + \beta \cdot \chi_p \cdot c_p \cdot \Psi_p^{k\,\prime} \cdot \left(v^k + dv^k - \bar{v}\right)\end{aligned} \tag{10}$$

with

$$\Psi_{d,f}^{k\,\prime} = \Psi_d' \left(\left(f(\boldsymbol{x} + \boldsymbol{w}^k + d\boldsymbol{w}^k) - f(\boldsymbol{x})\right)^2\right) \tag{11}$$

$$\Psi_{s1}^{k\,\prime} = \Psi_{s1}' \left(\left(\boldsymbol{r}_1^\top \nabla(u^k + du^k)\right)^2 + \left(\boldsymbol{r}_1^\top \nabla(v^k + dv^k)\right)^2\right) \tag{12}$$

$$\Psi_{s2}^{k\,\prime} = \Psi_{s1}' \left(\left(\boldsymbol{r}_2^\top \nabla(u^k + du^k)\right)^2 + \left(\boldsymbol{r}_2^\top \nabla(v^k + dv^k)\right)^2\right) \tag{13}$$

$$\Psi_{p}^{k\,\prime} = \Psi_{p}' \left(\left(u^k + du^k - \bar{u}\right)^2 + \left(v^k + dv^k - \bar{v}\right)^2\right) \tag{14}$$

1.5 Similarity Tensor

As a by-product, we also introduce a tensor notation for the similarity term. To understand this, we rewrite the expressions involved in (1) as vector products:

$$(u-\bar{u})^2 = \left(\begin{pmatrix}1\\0\\-\bar{u}\end{pmatrix}^\top \begin{pmatrix}u\\v\\1\end{pmatrix}\right)^2 = \begin{pmatrix}u\\v\\1\end{pmatrix}^\top \underbrace{\begin{pmatrix}1 & 0 & -\bar{u}\\ 0 & 0 & 0\\ -\bar{u} & 0 & \bar{u}^2\end{pmatrix}}_{S_u} \begin{pmatrix}u\\v\\1\end{pmatrix} \tag{15}$$

Similar, we can compute

$$S_v = \begin{pmatrix}0 & 0 & 0\\ 0 & 1 & -\bar{v}\\ 0 & -\bar{v} & \bar{v}^2\end{pmatrix} \tag{16}$$

Combining the two tensors via $S = S_u + S_v$, we can rewrite the similarity term as

$$\mathcal{E}_p = \chi_p \cdot c_p \cdot \Psi_p\left(\boldsymbol{w}^\top S \boldsymbol{w}\right) \tag{17}$$

Within the incremental formulation of the multi-scale scheme, we do not impose similarity of u and $\bar{u}$, but of du^k and $\bar{u}-u^k$; see corresponding Euler-Lagrange equations in (10)– (11). Thus, on each scale, we obtain $S^k = S_u^k + S_v^k$, with

$$S_u^k = \begin{pmatrix} 1 & 0 & u^k-\bar{u} \\ 0 & 0 & 0 \\ u^k-\bar{u} & 0 & (u^k-\bar{u})^2 \end{pmatrix} \quad S_v^k = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & v^k-\bar{v} \\ 0 & v^k-\bar{v} & (v^k-\bar{v})^2 \end{pmatrix} \tag{18}$$

1.6 Incremental Euler-Lagrange Equations with Similarity Tensor

Using the similarity tensor deduced above, we can rewrite the Euler-Lagrange equations (10)– (11) as

$$\begin{aligned} 0 = &-\alpha \cdot \operatorname{div}\left(\mathbf{D}^k \nabla u\right) + \Psi_{d,g}^{k\,\prime} \cdot \left(J_{11,g}^k du^k + J_{12,g}^k dv^k + J_{13,g}^k\right) \\ &+\lambda \cdot \Psi_{d,\nabla g}^{k\,\prime} \cdot \left(J_{11,\nabla g}^k du^k + J_{12,\nabla g}^k dv^k + J_{13,\nabla g}^k\right) \\ &+\beta \cdot \chi_p \cdot c_p \cdot \Psi_p^{k\,\prime} \cdot \left(S_{11}^k du^k + S_{12}^k dv^k + S_{13}^k\right) \end{aligned} \tag{19}$$

$$\begin{aligned} 0 = &-\alpha \cdot \operatorname{div}\left(\mathbf{D}^k \nabla v\right) + \Psi_{d,g}^{k\,\prime} \cdot \left(J_{12,g}^k du^k + J_{22,g}^k dv^k + J_{23,g}^k\right) \\ &+\lambda \cdot \Psi_{d,\nabla g}^{k\,\prime} \cdot \left(J_{12,\nabla g}^k du^k + J_{22,\nabla g}^k dv^k + J_{23,\nabla g}^k\right) \\ &+\beta \cdot \chi_p \cdot c_p \cdot \Psi_p^{k\,\prime} \cdot \left(S_{12}^k du^k + S_{22}^k dv^k + S_{23}^k\right) \end{aligned} \tag{20}$$

1.7 Differential Energy

Summarizing, the deduced equations can be shown to be the Euler-Lagrange equations of the energy

$$\begin{aligned} \mathcal{E}(du^k, dv^k) = \int_\Omega \Big[&\Psi_{d,g}^k \left(d\boldsymbol{w}^{k\,\top} J_g^k d\boldsymbol{w}^k\right) + \lambda \cdot \Psi_{d,\nabla g} \left(\boldsymbol{w}^{k\,\top} J_{\nabla_g}^k d\boldsymbol{w}^k\right) \\ &+ \alpha \cdot \Psi_{s_1} \left(\left(\boldsymbol{r}_1^\top \nabla (u^k + du^k)\right)^2 + \left(\boldsymbol{r}_1^\top \nabla (v^k + dv^k)\right)^2 \right) \\ &+ \alpha \cdot \Psi_{s_2} \left(\left(\boldsymbol{r}_2^\top \nabla (u^k + du^k)\right)^2 + \left(\boldsymbol{r}_2^\top \nabla (v^k + dv^k)\right)^2 \right) \\ &+ \beta \cdot \chi_p \cdot c_p \cdot \Psi_p \left(d\boldsymbol{w}^{k\,\top} S^k d\boldsymbol{w}^k\right) \Big] \; d\boldsymbol{x} \end{aligned} \tag{21}$$

2 Additional Results

In this section we show additional results of ALD-Flow. In particular, we show additional flow fields, intermediate steps of our adaptive sparsification strategy and further tables from the Middlebury benchmark [1]. In all cases we show the overlayed input frames, the results of the baseline method, the descriptor matches which are computed within the regions of interest and the final result of our method.

We start with Table 1 listing the average endpoint error (AEE) for our method and its baseline method for the sequences of the Middlebury training dataset. Note, that the corresponding table presenting the average angular error (AAE) is contained in our main paper.

Next, Figure 1 illustrates the different steps to compute the final result for frame 496 of the tennis sequence. Compared to our paper, we show even more intermediate steps concerning the selection of the different candidate sets. Analogously, we show these steps and our result for the Human Eva II sequence in Figure 2.

These illustrations are followed by Figure 3 in which all results for the Middlebury training sequences with ground truth are depicted. The results for the additional real-world sequences without ground truth are shown in Figure 4. Furthermore, in Figure 5, we present the results for those sequences of the Middlebury evaluation dataset that are considered for the rankings with respect to the AAE and the AEE. Figure 6 shows the additional sequences which are used for the interpolation error rankings.

Finally, we show the ranking of our method in the Middlebury benchmark with respect to the AEE, the interpolation error (IE) and the normalized interpolation error (NIE) in Figures 7, 8 and 9, respectively. As one can see, we achieve consistently good results, in particular with respect to the NIE. Note, that the corresponding table presenting the ranking with respect to the average angular error (AAE) is contained in our main paper.

References

1. Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., Szeliski, R.: A database and evaluation methodology for optical flow. *International Journal of Computer Vision* **92(1)** (2011) 1–31
2. Brox, T., Bruhn, A., Papenberg, N.,Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In *Proc. European Conference on Computer Vision*, Lecture Notes in Computer Science **3024** (2004) 25–36
3. Bruhn, A., Weickert, J.: Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proc. IEEE International Conference on Computer Vision* (2005) 749–755
4. Horn, B., Schunck, B.: Determining optical flow. *Artificial Intelligence* **17** (1981) 185–203
5. Zimmer, H., Bruhn, A., Weickert, J., Valgaerts, L., Salgado, A., Rosenhahn, B., Seidel, H.-P.: Complementary optic flow. In *Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science **5681** (2009) 207–220

Table 1. The AEE of ALD-Flow and its baseline method.

Method	Avg.	RubW.	Hydra.	Grove2	Grove3	Urban2	Urban3	Dime.	Venus
Baseline	0.218	**0.068**	**0.135**	**0.118**	0.521	0.214	0.336	**0.096**	0.256
ALD-Flow	**0.212**	0.069	**0.135**	**0.118**	**0.513**	**0.202**	**0.309**	**0.096**	**0.255**

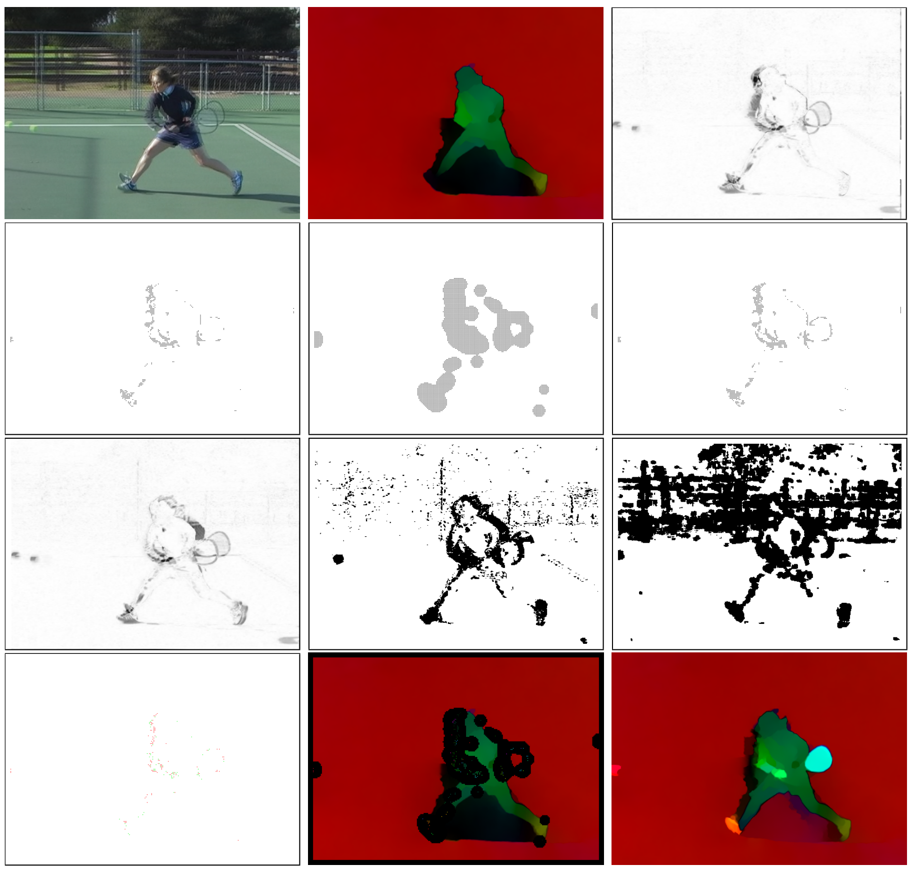

Fig. 1. Detailed illustration of flow estimation for frame 496 of the tennis sequence (best viewed in electronic version). **From left to right, top to bottom: (a)** Overlayed input frames, **(b)** baseline result, **(c)** energy of data term in frame 1, **(d)-(f)** double thresholding to compute candidate set s_1: first thresholding, regions of interest, second thresholding, **(g)** energy of data term in frame 2, **(h)** candidate set s_2. **(i)** relaxed candidate set s_{1b}, **(j)** distribution of HOG (red) and GB (green) descriptors, **(k)** feature matches within regions of interest, **(l)** ALD-flow result.

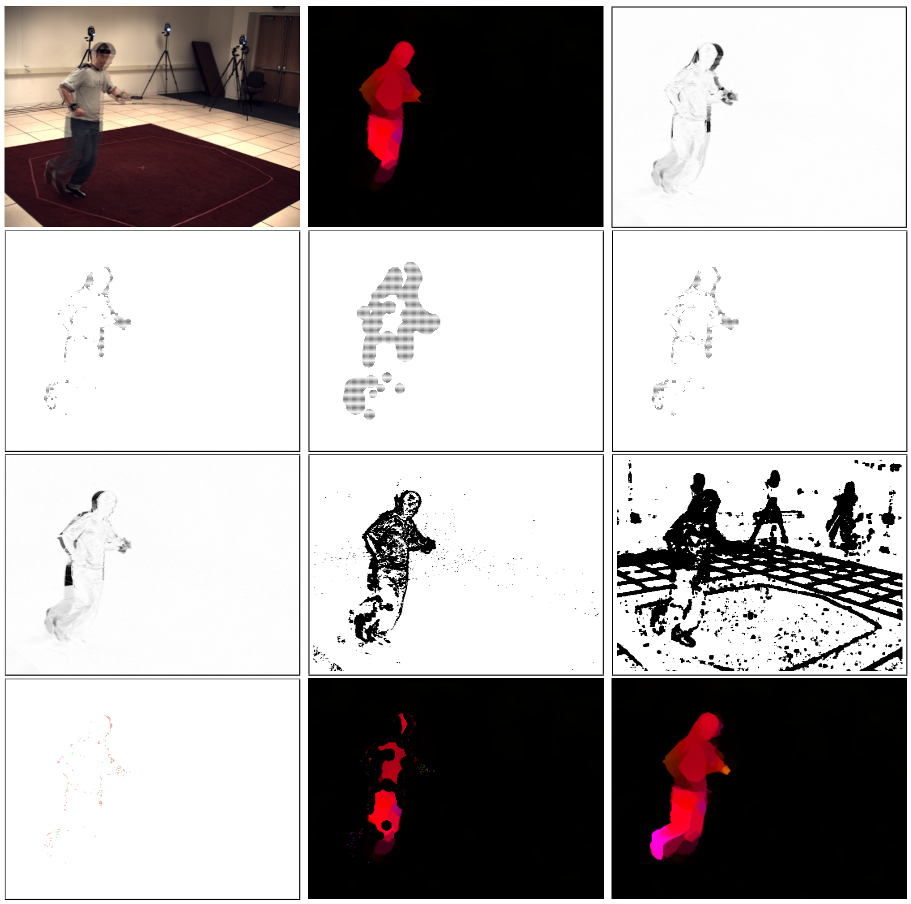

Fig. 2. Detailed illustration of flow estimation of the Human Eva II dataset (best viewed in electronic version). **From left to right, top to bottom: (a)** Overlayed input frames, **(b)** baseline result, **(c)** energy of data term in frame 1, **(d)-(f)** double thresholding to compute candidate set s_1: first thresholding, regions of interest, second thresholding, **(g)** energy of data term in frame 2, **(h)** candidate set s_2. **(i)** relaxed candidate set s_{1b}, **(j)** distribution of HOG (red) and GB (green) descriptors, **(k)** feature matches within regions of interest, **(l)** ALD-flow result.

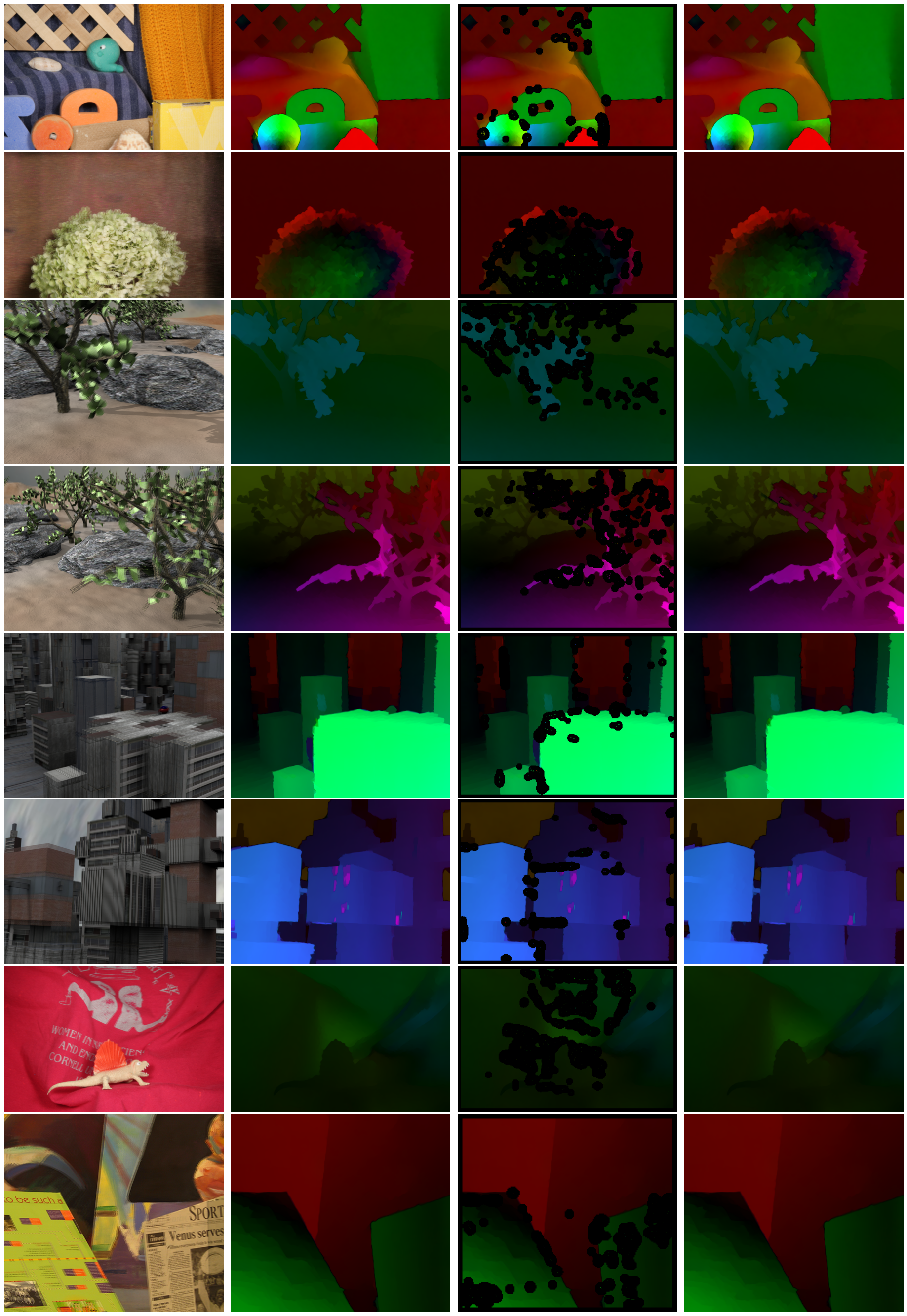

Fig. 3. Middlebury training sequences: RubberWhale, Hydrangea, Grove2, Grove3, Urban2, Urban3, Dimetrodon, Venus. **From left to right:** Overlayed input frames, baseline result, feature matches within regions of interest, ALD-flow result.

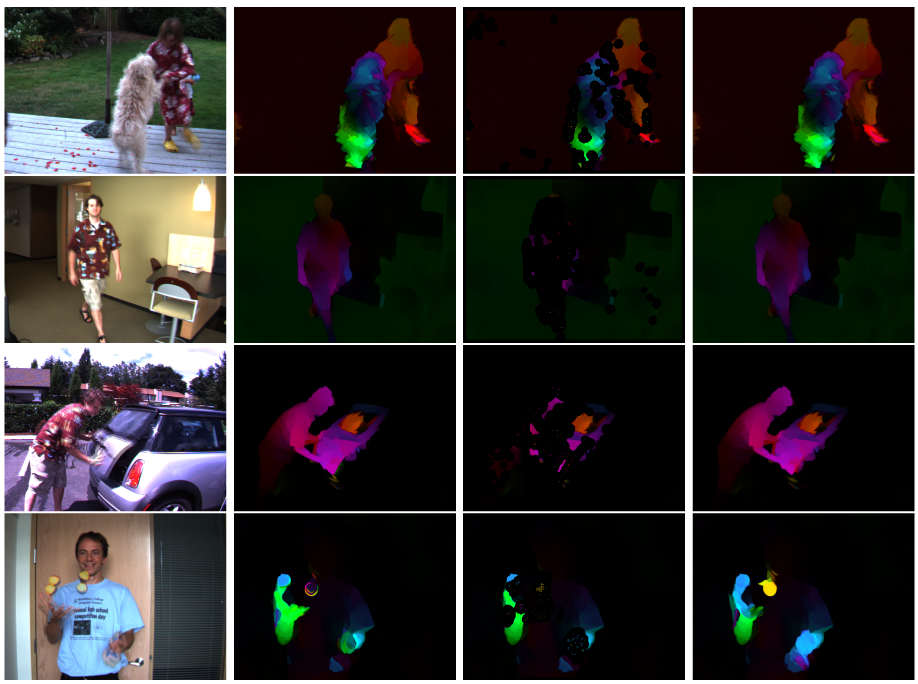

Fig. 4. Middlebury training sequences (real-world): DogDance, Walking, MiniCooper and Beanbags. **From left to right:** Overlayed input frames, baseline result, feature matches within regions of interest, ALD-flow result.

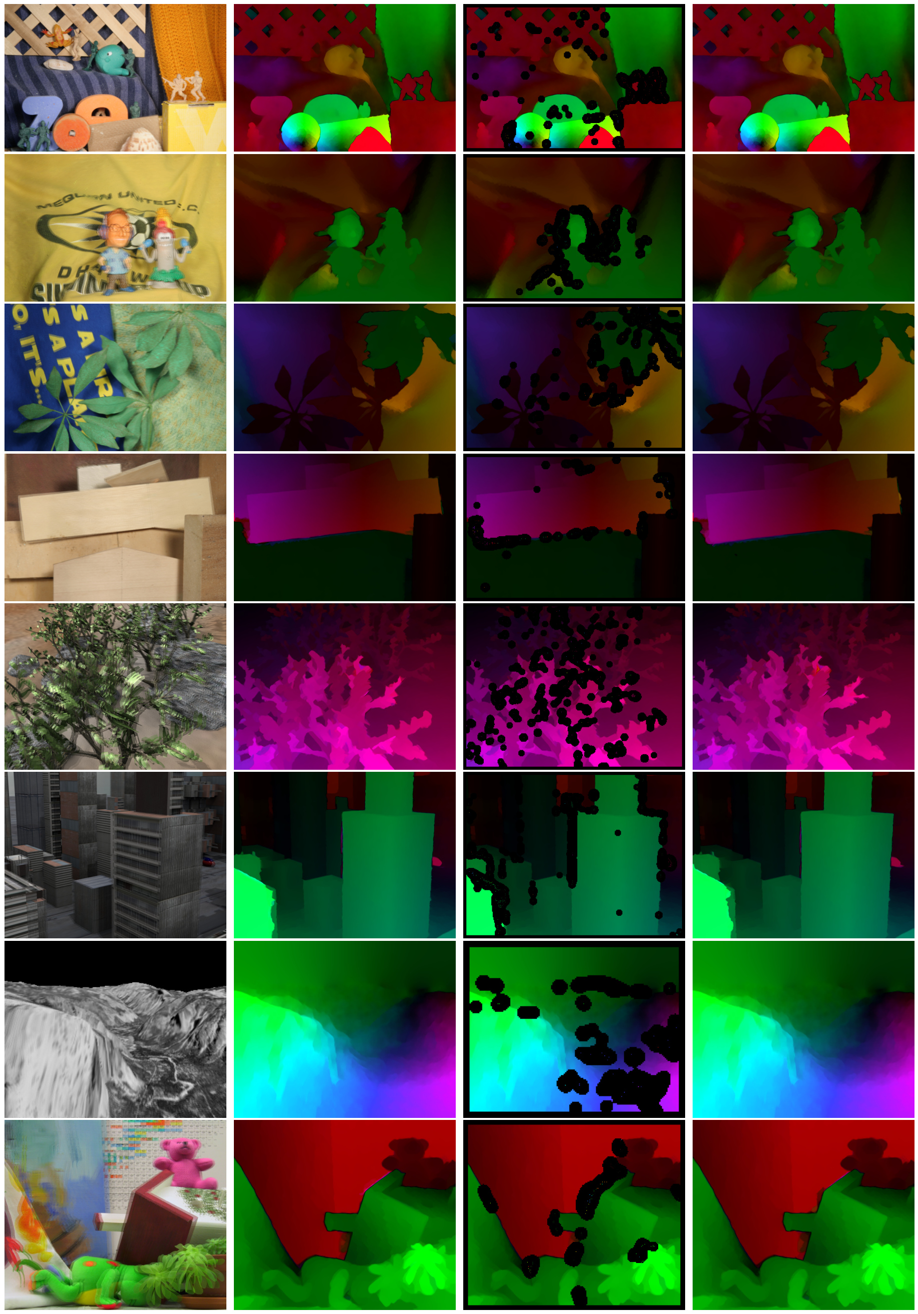

Fig. 5. Middlebury evaluation sequences: Army, Mequon, Schefflera, Wooden, Grove, Urban, Yosemite and Teddy. **From left to right:** Overlayed input frames, baseline result, feature matches within regions of interest, ALD-flow result.

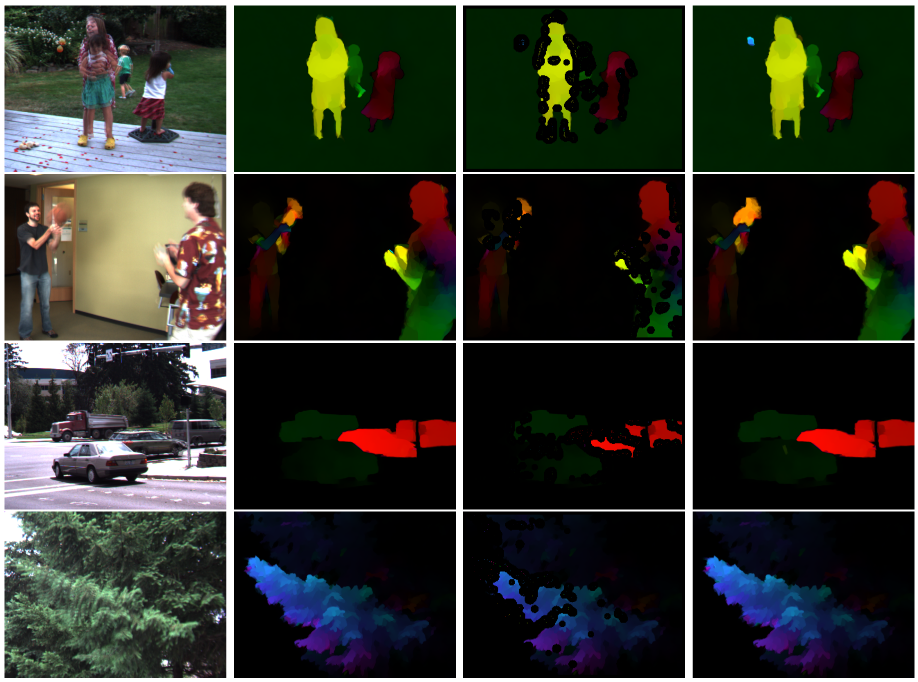

Fig. 6. Middlebury evaluation sequences (real-world): Backyard, Basketball, Dumptruck and Evergreen. **From left to right:** Overlayed input frames, baseline result, feature matches within regions of interest, ALD-flow result.

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
ADF [72]	8.6	0.08 4	0.22 5	**0.06** 1	0.18 3	0.62 8	0.14 8	0.29 14	0.71 17	0.17 7	0.16 15	0.91 25	**0.07** 1	0.69 12	1.03 12	0.47 8	0.43 11	0.91 3	0.28 6	0.12 15	0.12 6	0.20 18	**0.43** 1	**0.88** 1	0.63 5
IROF++ [62]	9.0	0.08 4	0.23 7	0.07 8	0.21 17	0.68 15	0.17 18	0.28 10	0.63 8	0.19 18	0.15 8	0.73 10	0.09 12	0.60 3	0.89 4	0.42 5	0.43 11	1.08 10	0.31 12	0.10 4	0.12 6	0.12 4	0.47 5	0.98 7	0.68 10
Layers++ [38]	9.2	0.08 4	0.21 2	0.07 8	0.19 7	0.56 3	0.17 18	**0.20** 1	**0.40** 1	0.18 12	**0.13** 1	0.58 2	**0.07** 1	**0.48** 1	**0.70** 1	0.33 2	0.47 19	1.01 6	0.33 18	0.15 37	0.14 30	0.24 31	0.46 3	**0.88** 1	0.72 13
MDP-Flow2 [40]	9.4	0.09 15	0.23 7	0.07 8	0.16 2	**0.52** 1	0.13 3	0.22 2	0.46 3	0.17 7	0.17 21	0.93 31	0.09 12	0.65 7	0.98 9	0.43 6	**0.29** 1	0.91 3	0.26 4	0.11 9	0.13 18	0.17 11	0.51 15	1.11 18	0.72 13
nLayers [61]	9.8	**0.07** 1	**0.19** 1	**0.06** 1	0.22 20	0.59 4	0.19 29	0.25 6	0.54 5	0.20 27	0.15 8	0.84 18	0.08 6	0.53 2	0.78 2	0.34 3	0.44 14	**0.84** 1	0.30 10	0.13 23	0.13 18	0.20 18	0.47 5	0.97 6	0.67 8
Sparse-NonSparse [59]	13.4	0.08 4	0.23 7	0.07 8	0.22 20	0.73 22	0.18 21	0.28 10	0.64 9	0.19 18	0.14 4	0.71 7	0.08 6	0.67 11	0.99 11	0.48 11	0.49 23	1.06 8	0.32 15	0.14 29	**0.11** 1	0.28 44	0.49 9	0.98 7	0.73 17
ALD-Flow [73]	13.5	**0.07** 1	0.21 2	**0.06** 1	0.19 7	0.64 13	0.13 3	0.30 17	0.73 18	0.15 2	0.17 21	0.92 29	**0.07** 1	0.78 21	1.14 21	0.59 21	0.33 3	1.30 22	**0.21** 1	0.12 15	0.12 6	0.28 44	0.54 18	1.19 21	0.73 17
COFM [63]	13.6	0.08 4	0.26 24	**0.06** 1	0.18 3	0.62 8	0.14 8	0.30 17	0.74 19	0.19 18	0.15 8	0.86 20	**0.07** 1	0.79 22	1.14 21	0.74 36	0.35 5	0.87 2	0.28 6	0.14 29	0.12 6	0.28 44	0.49 9	0.94 4	0.71 12
Efficient-NL [65]	13.9	0.08 4	0.22 5	**0.06** 1	0.23 30	0.73 22	0.18 21	0.32 22	0.75 20	0.18 12	0.14 4	0.72 9	0.08 6	0.60 3	0.88 3	0.43 6	0.57 32	1.11 14	0.35 24	0.14 29	0.13 18	0.25 34	0.48 7	0.90 3	0.63 5
⋮	⋮		⋮			⋮									⋮									⋮	
Complementary OF [21]	29.3	0.11 32	0.28 31	0.10 40	0.18 3	0.63 10	0.12 2	0.31 20	0.75 20	0.18 12	0.19 30	0.97 33	0.12 32	0.97 44	1.31 40	1.00 49	1.78 65	1.73 45	0.87 58	0.11 9	0.12 6	0.22 26	0.68 29	1.48 30	0.95 37
⋮	⋮		⋮			⋮									⋮									⋮	
Brox et al. [5]	38.5	0.11 32	0.32 40	0.11 46	0.27 40	0.93 41	0.22 42	0.39 29	0.94 28	0.24 38	0.24 40	1.25 45	0.13 35	1.10 56	1.39 51	1.43 63	0.89 44	1.77 49	0.55 41	0.10 4	0.13 18	0.11 2	0.91 44	1.83 48	1.13 47
⋮	⋮		⋮			⋮									⋮									⋮	
LDOF [28]	45.2	0.12 42	0.35 46	0.10 40	0.32 45	1.06 51	0.24 44	0.43 30	0.98 32	0.30 49	0.45 55	2.48 71	0.26 51	1.01 48	1.37 46	1.05 50	1.10 47	2.08 58	0.67 48	0.12 15	0.15 40	0.24 31	0.94 47	2.05 56	1.10 44

Fig. 7. Middlebury ranking with respect to average endpoint error (AEE).

Average interpolation error	avg. rank	Mequon (Hidden texture)			Schefflera (Hidden texture)			Urban (Synthetic)			Teddy (Stereo)			Backyard (High-speed camera)			Basketball (High-speed camera)			Dumptruck (High-speed camera)			Evergreen (High-speed camera)		
		im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT 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
MDP-Flow2 [40]	9.2	2.86 2	5.31 4	1.20 2	3.46 3	5.07 3	1.31 4	**3.49** 1	**5.34** 1	1.47 5	5.40 15	7.95 22	3.41 43	10.2 3	12.7 3	3.61 16	6.12 11	11.8 9	2.38 24	7.48 4	17.1 4	1.51 2	**7.32** 1	**11.4** 1	1.75 39
CBF [12]	13.4	**2.83** 1	5.20 2	1.23 26	3.97 28	5.79 24	1.56 30	3.62 5	5.47 3	1.60 12	5.21 2	**7.12** 1	3.29 25	10.1 2	12.6 2	3.62 20	5.97 5	11.5 5	2.31 11	7.76 19	17.8 19	1.61 18	7.60 8	11.9 8	1.76 45
Aniso. Huber-L1 [22]	15.9	2.95 9	5.44 8	1.24 32	4.42 46	6.27 46	1.67 43	3.79 15	5.70 9	1.50 6	5.31 6	7.42 7	3.24 22	11.1 23	14.0 28	3.61 16	5.91 3	11.4 3	**2.24** 1	7.60 9	17.3 6	1.51 2	7.62 10	11.9 8	1.73 24
CLG-TV [51]	16.3	2.94 8	5.45 9	1.25 38	4.26 39	6.17 37	1.60 35	3.68 10	5.73 11	1.73 17	5.36 8	7.41 6	3.32 36	11.1 23	14.0 28	3.57 6	5.88 2	11.3 2	2.26 2	7.58 7	17.0 3	1.57 15	7.75 15	12.1 15	1.72 19
IROF-TV [56]	17.0	3.07 21	5.91 27	1.23 26	3.71 14	5.47 12	1.40 14	3.70 11	6.27 26	1.58 11	5.25 3	7.60 10	3.17 9	11.0 18	13.9 20	4.47 61	6.37 25	12.4 27	2.30 9	7.79 21	17.9 21	**1.50** 1	7.63 11	11.9 8	**1.66** 1
LCM-flow [64]	17.3	2.86 2	**5.13** 1	1.25 38	3.94 26	5.87 26	1.64 38	3.87 19	6.60 35	1.79 19	5.37 10	7.29 4	3.30 30	**9.99** 1	**12.5** 1	3.56 3	6.12 11	11.8 9	2.26 2	7.76 19	17.7 17	1.68 33	7.58 7	11.8 5	1.80 60
IROF++ [62]	17.4	3.03 15	5.77 19	1.20 2	3.59 7	5.31 8	1.33 7	4.32 33	6.61 37	2.25 34	**5.06** 1	7.14 2	3.16 8	11.0 18	13.9 20	4.44 59	6.34 21	12.3 22	2.27 4	7.54 6	17.3 6	1.64 27	8.09 28	12.7 29	1.69 4
ALD-Flow [73]	18.5	3.28 47	6.45 50	1.24 32	3.81 19	5.73 22	1.41 17	3.62 5	6.28 27	1.35 2	5.58 24	8.39 38	3.04 3	10.8 9	13.5 9	4.15 47	5.96 4	11.4 3	2.29 7	**7.34** 1	**16.8** 1	1.51 2	8.25 36	12.9 33	1.70 7
Second-order prior [8]	19.6	2.91 7	5.39 7	1.24 32	4.26 39	6.21 40	1.56 30	3.82 17	6.34 29	1.62 13	5.39 13	7.68 13	3.04 3	11.1 23	13.9 20	3.59 9	6.14 13	11.9 14	2.31 11	7.61 10	17.4 11	1.63 25	7.90 20	12.4 22	1.78 49
⋮	⋮		⋮			⋮									⋮									⋮	
Brox et al. [5]	21.0	3.08 23	5.94 29	1.21 9	3.83 21	5.67 20	1.45 22	3.93 20	5.76 13	1.67 14	5.32 7	7.19 3	3.22 18	10.6 7	13.4 7	3.56 3	6.60 42	12.7 35	2.42 36	8.61 46	19.7 49	3.04 72	7.43 3	11.6 3	1.68 2
⋮	⋮		⋮			⋮									⋮									⋮	
LDOF [28]	27.5	3.03 15	5.66 14	1.28 50	4.06 30	5.53 15	2.40 72	4.32 33	6.43 32	2.00 25	5.45 17	7.56 8	3.60 54	10.2 3	12.7 3	3.59 9	6.39 26	12.4 27	2.29 7	8.36 40	19.4 45	2.21 61	7.57 5	11.8 5	1.86 65
⋮	⋮		⋮			⋮									⋮									⋮	
Complementary OF [21]	41.1	3.48 58	7.32 67	1.20 2	3.89 23	5.96 29	1.45 22	8.94 68	6.94 43	5.45 68	6.33 54	10.0 64	3.09 5	11.3 37	14.2 37	4.24 49	6.33 19	12.3 22	2.42 36	8.62 47	19.3 42	1.75 41	9.07 67	14.3 67	1.72 19

Fig. 8. Middlebury ranking with respect to interpolation error.

Average normalized interpolation error	avg. rank	Mequon (Hidden texture)			Schefflera (Hidden texture)			Urban (Synthetic)			Teddy (Stereo)			Backyard (High-speed camera)			Basketball (High-speed camera)			Dumptruck (High-speed camera)			Evergreen (High-speed camera)		
		im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT im1			im0 GT 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
MDP-Flow2 [40]	10.5	**0.58** 1	**0.71** 1	**0.64** 1	0.63 4	0.87 4	**0.59** 1	0.92 8	1.37 11	0.85 10	0.98 22	1.14 38	1.24 22	**0.98** 1	**0.95** 1	1.15 4	1.13 23	1.60 28	1.08 20	0.68 7	1.23 7	0.68 10	**0.75** 1	1.06 2	0.64 24
ALD-Flow [73]	15.1	0.62 20	0.81 19	0.66 32	0.70 19	0.99 20	0.62 19	0.87 2	1.28 5	0.65 2	0.94 3	1.01 4	1.21 5	1.09 30	1.12 30	1.54 48	1.03 2	**1.24** 1	1.07 16	0.64 2	1.12 2	0.65 2	0.97 39	1.44 38	0.63 2
CLG-TV [51]	18.7	0.63 33	0.86 34	0.66 32	0.81 41	1.12 40	0.66 33	0.96 12	1.43 13	0.96 21	0.97 15	1.03 8	1.25 25	1.06 22	1.08 24	1.15 4	**1.02** 1	1.25 2	1.04 2	**0.63** 1	**1.09** 1	0.66 4	0.97 39	1.45 40	0.63 2
ADF [72]	18.8	0.59 2	0.73 2	**0.64** 1	0.68 15	0.97 15	0.62 19	0.89 4	1.29 6	0.79 7	0.93 2	0.99 2	1.20 2	1.19 56	1.29 56	1.69 61	1.04 3	1.33 4	1.04 2	0.84 39	1.71 40	0.76 35	0.88 19	1.28 19	0.65 41
LCM-flow [64]	19.1	0.62 20	0.80 16	0.66 32	0.77 31	1.07 31	0.71 48	1.03 18	1.70 31	0.91 18	1.01 30	1.07 17	1.27 31	0.99 2	**0.95** 1	1.16 18	1.07 9	1.43 11	1.04 2	0.67 6	1.20 6	0.71 18	0.84 11	1.21 11	0.65 41
Aniso. Huber-L1 [22]	19.7	0.62 20	0.80 16	0.66 32	0.84 45	1.13 42	0.66 33	1.03 18	1.44 14	0.93 20	0.97 15	1.03 8	1.26 30	1.06 22	1.09 25	1.15 4	1.08 12	1.46 14	**1.03** 1	0.64 2	1.12 2	0.66 4	0.99 44	1.48 47	0.63 2
IROF++ [62]	19.7	0.59 2	0.74 3	**0.64** 1	0.65 8	0.89 7	**0.59** 1	1.15 29	1.71 32	1.17 30	**0.92** 1	**0.96** 1	1.21 5	1.17 48	1.26 48	1.69 61	1.11 16	1.54 17	1.04 2	0.68 7	1.23 7	0.70 14	1.07 65	1.62 66	0.63 2
IROF-TV [56]	20.8	0.62 20	0.84 30	0.65 13	0.67 13	0.92 13	0.60 6	0.92 8	1.49 20	0.79 7	0.94 3	1.02 6	1.22 15	1.18 51	1.28 51	1.70 67	1.12 19	1.58 23	1.05 8	0.79 30	1.57 32	0.70 14	0.85 12	1.24 13	0.64 24
Second-order prior [8]	21.0	0.61 9	0.78 10	0.66 32	0.80 40	1.11 37	0.64 28	1.05 22	1.85 41	0.99 23	0.96 11	1.04 12	1.21 5	1.05 18	1.07 20	1.15 4	1.05 5	1.38 6	1.05 8	0.69 10	1.28 11	0.65 2	1.00 48	1.50 50	0.66 53
⋮	⋮		⋮			⋮									⋮									⋮	
Brox et al. [5]	28.2	0.67 53	1.04 63	0.65 13	0.72 25	1.02 27	0.63 23	0.96 12	1.34 8	0.83 9	0.98 22	0.99 2	1.24 22	1.02 6	1.02 8	1.15 4	1.20 46	1.78 49	1.11 37	1.67 72	3.86 72	2.48 72	0.86 15	1.26 15	**0.62** 1
⋮	⋮		⋮			⋮									⋮									⋮	
LDOF [28]	29.4	0.66 46	0.94 45	0.67 45	0.79 37	0.99 20	0.82 66	1.15 29	1.37 11	1.14 29	0.98 22	1.08 21	1.24 22	1.00 4	0.98 4	1.15 4	1.06 7	1.39 7	1.04 2	1.14 66	2.51 67	1.27 70	0.83 9	1.19 9	0.67 64
⋮	⋮		⋮			⋮									⋮									⋮	
Complementary OF [21]	38.0	0.66 46	1.03 61	**0.64** 1	0.70 19	1.01 25	0.63 23	3.10 68	2.52 57	3.34 68	0.98 22	1.13 35	1.22 15	1.16 43	1.25 45	1.59 53	1.13 23	1.59 24	1.10 30	0.93 54	1.87 51	0.97 63	0.94 29	1.40 32	0.64 24

Fig. 9. Middlebury ranking with respect to normalized interpolation error.