GUIDING OPTICAL FLOW ESTIMATION USING SUPERPIXELS

Theodosios Gkamas *Christophoros Nikou*

University of Ioannina
Department of Computer Science
45110 Ioannina, Greece
{tgkamas, cnikou}@cs.uoi.gr

ABSTRACT

In this paper, we show how the segmentation of an image into superpixels may be used as preprocessing paradigm to improve the accuracy of the optical flow estimation in an image sequence. Superpixels play the role of accurate support masks for the integration of the optical flow equation. We employ a variation of a recently proposed optical flow algorithm relying on local image properties that are taken into account only if the involved pixels belong to the same image segment. Experimental results show that the proposed optical flow estimation scheme significantly improves the accuracy of the estimated motion field with respect to other standard methods.

***Index Terms*—** Optical flow, image segmentation, superpixels.

1. INTRODUCTION

The estimation of optical flow is one of the fundamental problems in computer vision as it provides the motion of brightness patterns in an image sequence. From a computational point of view, there are two main families of methods for optical flow computation. The first category consists of local techniques, relying on an isotropic coarse-to-fine image warping, having as their major representative the Lucas-Kanade algorithm [1]. A Gaussian or rectangular window adapted in scale but being isotropic controls a local neighborhood and jointly with a pyramidal implementation is capable of extending motion estimates from corners to edges and the interior of regions. This method and its variants are still among the most popular for flow and feature tracking. The second family of optical flow methods are the global or variational techniques, relying on an energy minimization framework, with their main representative being the Horn-Schunck method [2], which optimizes a cost function using both brightness constancy and global flow smoothness and has also led to many variants of the basic idea.

The computation of optical flow requires spatial integration because local signals are noisy and suffer from the well-known aperture problem [2]. This integration is a non-trivial task and yields a grouping question: the association of a set of pixels with a motion model. Methods employing isotropic neighborhoods are faced with the dilemma of small neighborhoods containing little information or large regions including motion or object boundaries. Apart from [2], among the first attempts to handle this issue is the work of Nagel and Enkelmann [3], where smoothing is discouraged at areas at regions with rich first and second spatial derivative information. More specifically, the *oriented smoothness* constraint is introduced which restricts variations of the displacement vector field only in directions with small or no variation of image intensities.

The representation of different motions with layers [4, 5] is another approach to handle the discontinuity problem. Also, simultaneous motion segmentation and optical flow computation with robust estimators [6, 7] was proposed to alleviate the effect of integrating outlying data. Small sized segments were employed in [8] but the approach has the shortcoming that small segments cannot faithfully represent the structure of natural scenes and they weaken the advantage provided by the involved segmentation. Also, the small size of the segments may result in erroneous flow estimation due to the aperture problem. The method presented in [9] employs both color and motion segmentation, obtained by using the mean shift algorithm, and an affine motion is estimated at the segmented regions. Finally, a combination of pixel grouping and pairwise pixel affinities, taking into account possible intervening contours, avoids motion propagation across boundaries in [10].

In this work, we propose an optical flow estimation scheme relying on image segmentation by superpixels [11]. The computed superpixels represent a robust oversegmentation of the image into roughly equally sized and similar in shape segments which preserve texture and color boundaries. Therefore, they play the role of accurate support masks for the integration of optical flow. To this end, we employ a variant of the recently proposed joint Lucas-Kanade (JLK) tracker [12], initially conceived for sparse feature tracking, which yields dense flow field estimates. Numerical experiments were performed on challenging sequences of the Middlebury [13] image data base which underpinned the performance of the proposed scheme.

978-1-4577-0274-7/11/$26.00 ©2011 IEEE

2. SEGMENTATION-BASED OPTICAL FLOW

The fundamental assumption for the estimation of both dense and sparse optical flow is the brightness constancy constraint whose approximation by a Taylor series expansion provides the well-known equation at a given image pixel:

$$I_x u + I_y v + I_t = 0 \tag{1}$$

where I_x and I_y are the partial image derivatives along the horizontal and vertical directions respectively, $\mathbf{u} = (u, v)^T$ is the motion vector to be estimated and I_t is the temporal image difference between consecutive frames.

In [12], the authors, based on a framework proposed by Bruhn *et al.* [14], presented a combination of Lucas-Kanade [1] and Horn-Schunck [2] energy functionals for sparse feature tracking which resulted in the so called joint Lucas-Kanade (JLK) scheme. The functional to be minimized has the following form:

$$E_{JLK} = \sum_{i=1}^{N} E_D(u_i, v_i) + \lambda_i E_S(u_i, v_i) \tag{2}$$

where *N* is the number of image features (corners or edges) and λ_i is a regularization parameter. The data term E_D is the optical flow constraint:

$$E_D(u_i, v_i) = K_\rho * \left((I_x u_i + I_y v_i + I_t)^2 \right) \tag{3}$$

where K_ρ is a suitable convolution kernel whose size determines the number of neighboring pixels to be aggregated and assigns appropriate weights to the pixels inside the window. Also, the smoothness term:

$$E_S(u_i, v_i) = (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 \tag{4}$$

controls the deviation of the displacement $\mathbf{u}_i = (u_i, v_i)^T$ of the i-th feature with respect to the expected displacement $\hat{\mathbf{u}}_i = (\hat{u}_i, \hat{v}_i)^T$ at the same feature.

In equation (2), the energy of the i-th feature is determined by the matching of the motion vector $(u_i, v_i)^T$ to the local image data, as well as by the deviation of this motion vector from the expected displacement $(\hat{u}_i, \hat{v}_i)^T$. Note that the expected displacement is not necessarily required to be the average of the neighboring displacements.

Differentiating E_{JLK} in (2) with respect to the motion vectors $(u_i, v_i)^T, i = 1, \ldots, N$, and setting the derivatives to zero, yields a large 2*N*x2*N* sparse matrix equation, whose $(2i-1)$-th and $(2i)$-th rows are:

$$\mathbf{Z}_i \mathbf{u}_i = \mathbf{e}_i \tag{5}$$

where

$$\mathbf{Z}_i = \begin{bmatrix} \lambda_i + K_\rho * (I_x I_x) & K_\rho * (I_x I_y) \\ K_\rho * (I_x I_y) & \lambda_i + K_\rho * (I_y I_y) \end{bmatrix} \tag{6}$$

and

$$\mathbf{e}_i = \begin{bmatrix} \lambda_i \hat{u}_i - K_\rho * (I_x I_t) \\ \lambda_i \hat{v}_i - K_\rho * (I_y I_t) \end{bmatrix} \tag{7}$$

and the image derivatives I_x, I_y and I_t are computed at the i-th location. This sparse system of equations may be solved using Jacobi iterations:

$$u_i^{(k+1)} = \hat{u}_i^{(k)} - \frac{J_{xx}\hat{u}_i^{(k)} + J_{xy}\hat{v}_i^{(k)} + J_{xt}}{\lambda_i + J_{xx} + J_{yy}} \tag{8}$$

$$v_i^{(k+1)} = \hat{v}_i^{(k)} - \frac{J_{xy}\hat{u}_i^{(k)} + J_{yy}\hat{v}_i^{(k)} + J_{yt}}{\lambda_i + J_{xx} + J_{yy}} \tag{9}$$

where $J_{xx} = K_\rho * \left(I_x^2\right)$, $J_{xy} = K_\rho * (I_x I_y)$, $J_{xt} = K_\rho * (I_x I_t)$, $J_{yy} = K_\rho * \left(I_y^2\right)$ and $J_{yt} = K_\rho * (I_y I_t)$ are all computed at the i-th pixel location.

According to [12], the expected motion displacement $\hat{\mathbf{u}}_i$ of a given feature is predicted by fitting an affine motion model to the displacements of the surrounding features, which are inversely weighted according to their distance to the central feature. The authors use a Gaussian weighting function on the distance with $\sigma = 10$ pixels. In our case, as we wish to employ the method for dense optical flow, we also use a kernel K_ρ for the estimation of the expected motion. Hence, K_ρ is a 19×19 Gaussian kernel of width $\sigma = 4$ (an isotropic average kernel could also be employed). This leads to a Horn-Schunck scheme [2] with the modification of a weighted estimation (using kernel K_ρ) of both the expected values of the motion vectors $\hat{\mathbf{u}}_i = (\hat{u}_i, \hat{v}_i)^T$ and the quantities J_{xx}, J_{yy}, J_{xy}, J_{xt} and J_{yt} in (8) and (9).

Also, as feature tracking may involve occlusions and lighting discontinuities, the authors choose to set the same value for the regularization parameter λ_i at each feature. In our case, we perform a segmentation of the first frame of the sequence to a number of superpixels [15, 11] which respects strong edges in an image. Such low level segmentation can be effectively computed using normalized cuts [16], which is a spectral clustering algorithm, with a conservative homogeneity threshold. In this work, we have employed a publicly available superpixel code [11]. As we consider only local connections into the affinity matrix, the superpixels are roughly homogeneous in size and shape. Representative segmentations into superpixels are shown in figure 1. Notice that the provided oversegmentation preserves object boundaries and has the tendency to construct homogeneous segments in both color and texture.

The assumption is that pixels belonging to the same superpixel should be considered in the integration window as they are probable to follow the same motion model. On the other hand, if a pixel belongs to a different superpixel with respect to the i-th pixel (the center of the integration window whose motion vector is to be updated) then it is more probable to have a different motion vector and it should not be accounted.

The idea is analogous with the adaptive smoothing mechanism generally used to determine the value of parameter λ_i

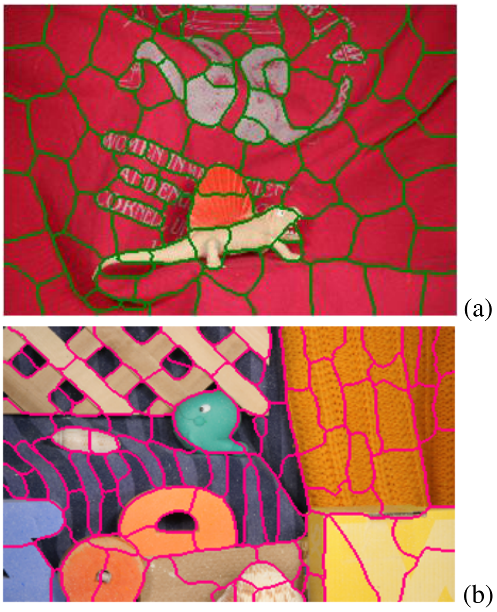

Fig. 1. Image oversegmentation into superpixels. Representative frames of (a) *Dimetrodon* and (b) *RubberWhale* sequences [13].

in (2). For example, in [3], if the neighborhood of the central pixel is rich in first and second derivative information this is an indication that the window is probable to integrate region boundaries and consequently pixels following different motions and the smoothness term is canceled. In the proposed scheme, we move one step ahead and perform a segmentation of the first frame of the sequence which is then used as a mask for the pixels to be integrated in the optical flow computation. The segmentation map switches on or off the smoothness term for each pixel in the integration window depending on whether the pixel in question belongs to the superpixel containing the central pixel.

3. EXPERIMENTAL RESULTS

Many optical flow methods have been proposed in the literature. As our approach is an extension of the joint Lucas-Kanade (JLK) method [12] we compared it to this algorithm. We have also included the well-known and established algorithm of Nagel and Enkelmann (NE) [3] which is based on the variational solution proposed by Horn and Schunck [2] associated with a selective smoothing. To visualize the motion vectors we adopt the color coding of figure 2.

The proposed method was tested on image sequences including both synthetic and real scenes. We have applied our method to the *Yosemite* sequence and its version with cloudy sky, called *Yosemite with clouds*. We have also used sequences obtained from the Middlebury database [13], such as the *Dimetrodon* and the *RubberWhale* sequences which contain nonrigid motions and large areas with little (hidden or not) texture.

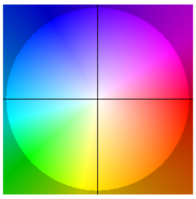

Fig. 2. The optical flow field color-coding. Smaller vectors are lighter and color represents the direction.

In order to evaluate the performance of the method, two performance measures were computed. The first measure of performance that we use in the comparison is the average angular error (AAE) [17]. This is the most common measure of performance for optical flow [13]. Let $\mathbf{v}_0 = (u_0\ ,\ v_0)$ be the correct velocity and $\mathbf{v}_1 = (u_1\ ,\ v_1)$ be the estimated velocity. The angular error (AE) between these two vectors is

$$\psi_{AE} = arccos\left(\vec{v_0} \cdot \vec{v_1}\right) \tag{10}$$

where $\vec{v_0}$, $\vec{v_1}$ are the 3D normalized representations of $\mathbf{v}_0$, $\mathbf{v}_1$, respectively and defined as

$$\vec{v_0} = \frac{1}{\sqrt{{u_0}^2 + {v_0}^2 + 1}} \left(u_0, v_0, 1\right) \tag{11}$$

$$\vec{v_1} = \frac{1}{\sqrt{{u_1}^2 + {v_1}^2 + 1}} \left(u_1, v_1, 1\right) \tag{12}$$

The AAE is then obtained by calculating the average of all angular errors between correct and estimated velocities in the optical flow. However, it can be seen from Eq. (10) that errors in regions of large flows are penalized less in AE than errors in regions of small flows [13]. Thus, one needs to be cautious when using the AAE metric as estimates with the same error magnitude may result in significantly different angular error values.

Another error metric is the normalized magnitude of the vector difference between the correct and estimated flow vectors [18]. The magnitude of the correct velocity is used as a normalization factor. The magnitude of difference error is defined as

$$E_M = \begin{cases} \frac{\|\mathbf{v}_0 - \mathbf{v}_1\|}{\|\mathbf{v}_0\|} & \text{, if } \|\mathbf{v}_0\| \geq T \\ \left|\frac{\|\mathbf{v}_1\| - T}{T}\right| & \text{, if } \|\mathbf{v}_0\| < T \text{ and } \|\mathbf{v}_1\| \geq T \\ 0 & \text{, if } \|\mathbf{v}_0\| < T \text{ and } \|\mathbf{v}_1\| < T \end{cases} \tag{13}$$

where T is a threshold, whose purpose is to ignore vector norms with values less than T. The algorithm is not expected to reliably produce accurate flow vectors in areas where the actual flow magnitude is less than the value of parameter T [18]. We used $T = 0.35$ in all of our experiments. The average magnitude of difference error (AME) is then calculated as the average of the normalized magnitude of difference errors.

The numerical results are summarized in Table 1, where it may be observed that the method proposed in this paper provides better accuracy with regard to the other methods. More specifically, our algorithm largely outperforms both the joint Lucas-Kanade method (JLK) and the selective smoothing scheme of Nagel and Enkelmann (NE). Notice that the accuracy of the JLK algorithm depends strongly on a Lucas-Kanade scheme. Therefore, we may conclude that JLK may perform better for sparse optical flow applied to features [12] than to dense flow estimation.

Representative results are illustrated in figure 3. As it may be seen, the segmentation-based optical flow estimation scheme provides correct estimates and simultaneously preserves the boundary information in the flow field. This is more pronounced in the *RubberWhale* sequence, where the solution of both the JLK and NE algorithms have artifacts in textured areas. For instance, JLK provides different angles for the motion vectors of the x-shaped area at the top left part of the figure and NE computes a relatively less smooth motion field at the top right part of the figure (light blue area). Also, the JLK scheme failed to compute a smooth estimate of the motion vectors of the *Dimetrodon* sequence, where NE performed better but not as well as the proposed scheme. Let us finally notice that, in the *Yosemite with clouds* sequence, all of the methods give a relatively non smooth solution for motion field of the clouds. Visually, JLK is smoother but the magnitudes of the estimated displacement vectors have very high values (e.g. the intense red area in the clouds) which leads to a decrease in the overall performance.

Furthermore, the above comments are also confirmed by the cumulative histograms for the AAE and AME for all of the compared algorithms, shown in figure 4. A point on the curve represents the percentage of optical flow errors that are less or equal than the value of the respective error on the horizontal axis. The higher the curve the better is the performance of the method. An ideal performance would provide a curve parallel to the horizontal axis, meaning that all of the errors are zero. Observe that the proposed algorithm provides rapidly increasing curves in all cases, which means that the majority of the errors are of relatively low amplitude.

The execution time of the proposed algorithm depends mainly on the number of superpixels and the image size. The results presented above were obtained by a segmentation of the image to 40 superpixels and a 19×19 window representing the involved neighborhood. Generally, experiments using some decades of superpixels need less than a minute on a standard PC running MATLAB.

4. CONCLUSION

The optical flow estimation method proposed in this paper relies on the segmentation of the image into a number of superpixels which guide the optical flow integration scheme. It was demonstrated that the idea is consistent and the proposed scheme improves the accuracy of standard optical flow methods. A perspective of this study is to estimate the number of superpixels as well as the size of the involved integration window automatically from the image data. Moreover, including more flexible motion models, such as affine or projective motions, would increase the estimation accuracy.

5. REFERENCES

[1] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th International Joint Conference on Artificial Intellegence, (IJCAI)*, 1981, pp. 674–679.

[2] B. Horn and B. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[3] H. Nagel and W. Enkelmann, “An investigation of smoothness constraints for estimation of displacement vector fields from image sequences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 565–593, 1986.

[4] J. Wang and E. Adelson, “Representing moving images with layers,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.

[5] H. Yalcin, M. Black, and R. Fablet, “The dense estimation of motion and appearance in layers,” in *2^{nd} IEEE Workshop on Image and Video Registration (IVR)*, 2004.

[6] M. 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.

[7] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, “Anisotropic Huber-L^1 optical flow,” in *British Machine Vision Conference (BMVC)*, 2009.

[8] C. Zitnick, N. Jojic, and S. Kang, “Consistent segmentation for optical flow estimation,” in *International Conference on Computer Vision, (ICCV)*, 2005, vol. 2, pp. 1308–1315.

[9] L. Xu, J. Chen, and J. Jia, “A segmentation based variational model for accurate optical flow estimation,” in 10^{th} *European Conference on Computer Vision (ECCV)*, 2008.

[10] X. Ren, “Local grouping for optical flow,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[11] G. Mori, “Guiding model search using segmentation,” in *10^{h} IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, vol. 2, pp. 1417–1423.

Table 1. Optical flow errors for the compared methods.

Method	*Yosemite*		*Yosemite with clouds*		*Dimetrodon*		*RubberWhale*	
	AAE	AME	AAE	AME	AAE	AME	AAE	AME
JLK [12]	7.97°	0.17	16.69°	0.35	33.14°	0.65	18.44°	0.43
NE [3]	9.15°	0.19	19.78°	0.47	17.58°	0.38	11.87°	0.29
Proposed method	**3.79°**	**0.09**	**11.86°**	**0.30**	**6.24°**	**0.18**	**8.17°**	**0.21**

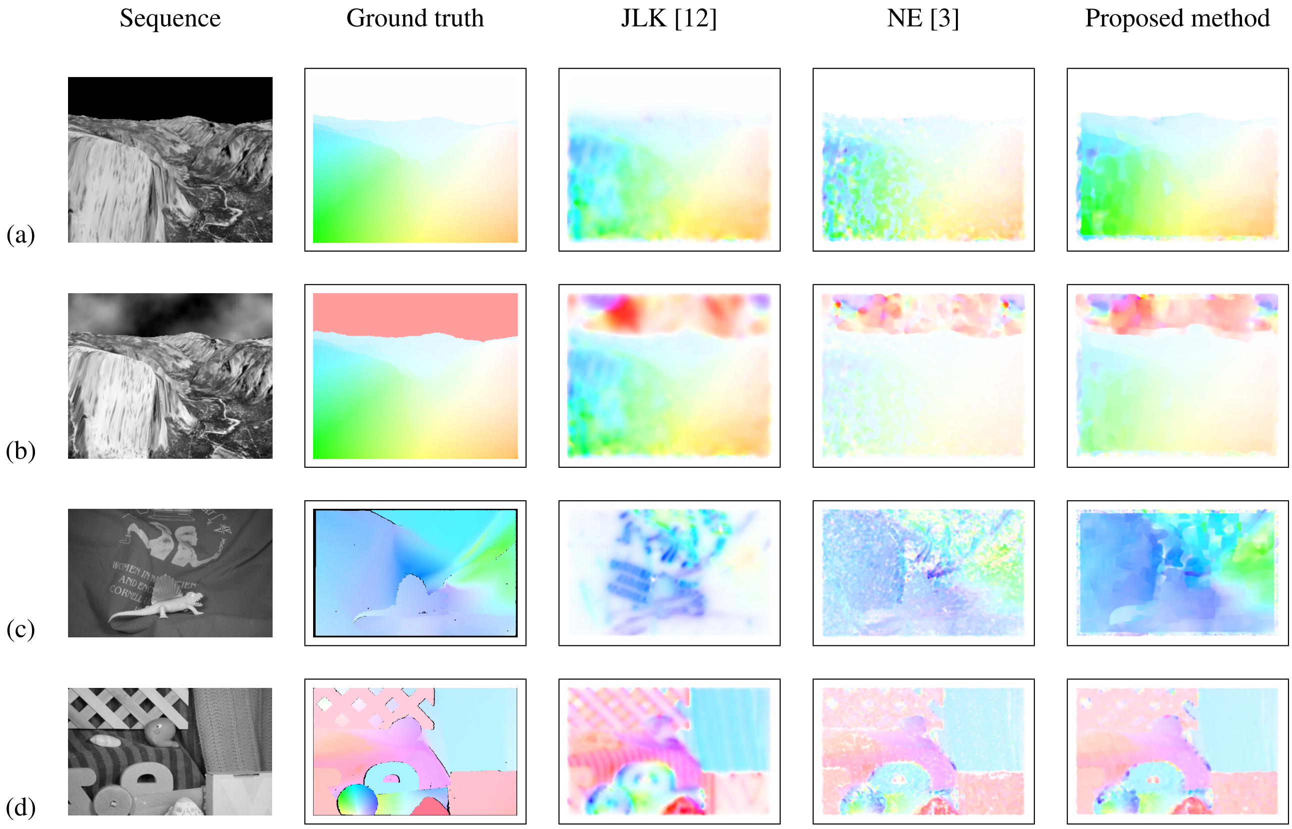

Fig. 3. Representative optical flow results, following the coding in fig. 2, for the sequences: (a) *Yosemite* , (b) *Yosemite with clouds*, (c) *Dimetrodon* and (d) *RubberWhale*.

[12] S. Birchfield and S. Pundlik, "Joint tracking of features and edges," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[13] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proceedings of the International Conference of Computer Vision, (ICCV)*, 2007, pp. 1–8.

[14] A. Bruhn, J. Weickert, and C. Schnorr, "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.

[15] X. Ren and J. Malik, "Learning a classification model for segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 2003, vol. 1, pp. 10–17.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[17] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision, (IJCV)*, vol. 12, no. 1, pp. 43–77, 1994.

[18] B. McCane, K. Novins, D. Crannitch, and B. Galvin, "On benchmarking optical flow," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 126–143, 2001.

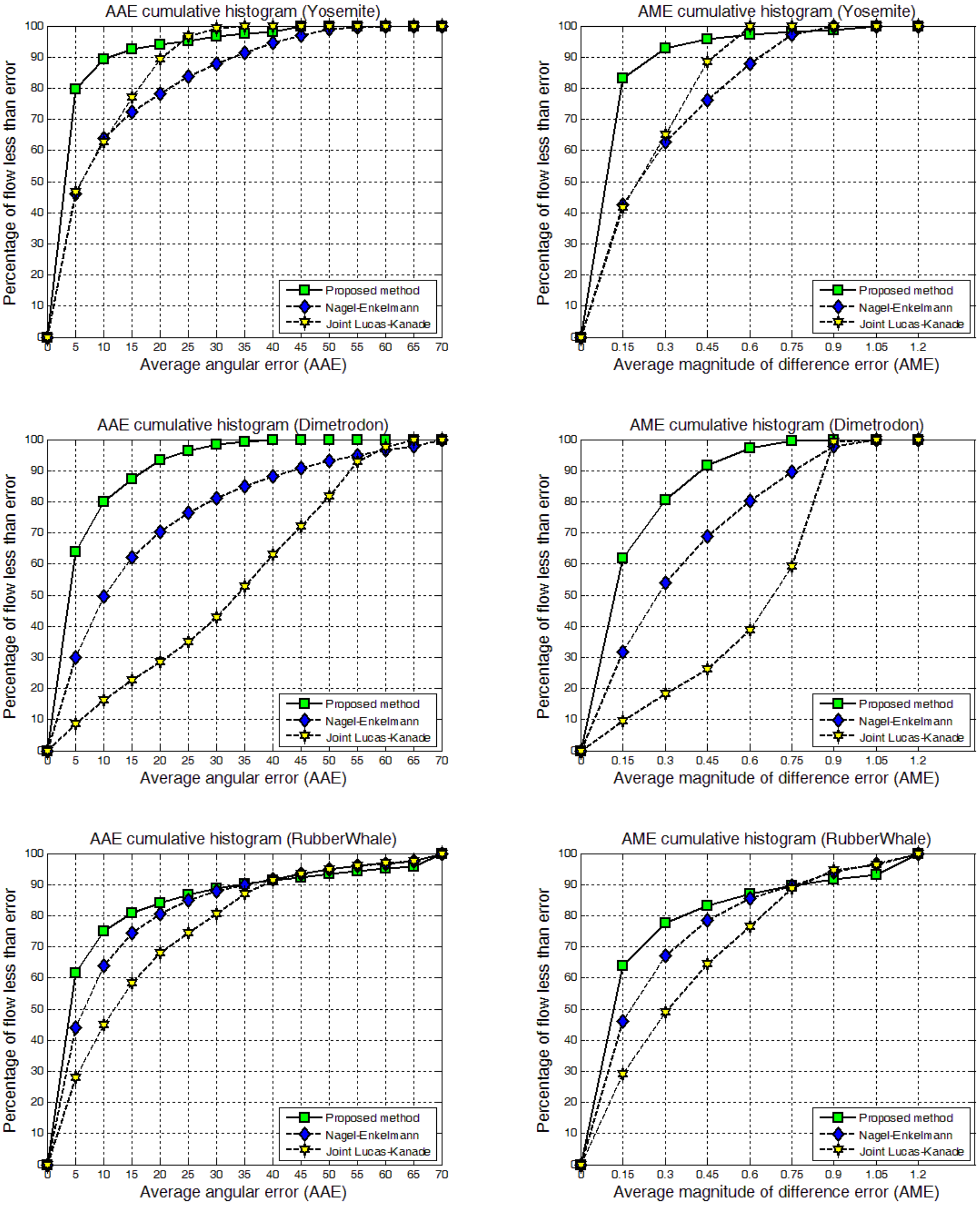

Fig. 4. Performances of the compared algorithms on the test sequences. From top to bottom: *Yosemite*, *Dimetrodon* and *RubberWhale*. Cumulative histograms showing the percentage of the optical flow errors which are lower than a given value (represented along the horizontal axis) for the AAE (left column) and the AME (right column).