Discontinuity-Preserving Optical Flow Computation by a Dynamic Overdetermined System

Yan Niu, Anthony Dick, Michael Brooks
School of Computer Science
University of Adelaide
North Terrace, Adelaide 5005, Australia
yan.niu@adelaide.edu.au

Abstract

This paper presents an optical flow estimation technique that improves the accuracy of existing methods in the problematic case of motion discontinuity. An initial flow estimate at a pixel is calculated from selected "reliable" pixels in the spatial neighbourhood. This initial estimate is then used to distinguish smooth and discontinuous regions. The flow estimate in a smooth region is refined using data from the temporal neighbourhood. Flow in a discontinuous region is estimated by reasoning about the local motion boundary, which can result in accurate estimation even when the optical flow constraint does not hold. By carefully structuring this computation, the method runs at the same speed as existing methods in the literature, while preliminary experiments indicate it can produce more accurate results near motion boundaries.

1. Introduction

Computing optical flow, the apparent motion of an image's brightness pattern, is a fundamental problem in computer vision. It provides crucial information for registration, tracking, stereo matching and many other motion related tasks. Differential techniques, which exploit the spatial and temporal variation of image intensity, have been intensely investigated for optical flow computation.

The common starting point of differential techniques is the *Optical Flow Constraint* (OFC), which assumes that the brightness of the moving objects remains constant. This constraint alone is not sufficient to recover a 2-dimensional velocity vector for each pixel. Therefore, the main concern of differential techniques is to integrate other appropriate constraints. The pioneering methods to solve the problem are Lucas-Kanade's constant flow [8] and Horn-Schunck's variation minimization [7], which both assume smoothness of the flow field.

In [7] by Horn-Schunck, it is suggested that each velocity component's gradient should have an l^2 norm near to zero. The trade-off between this smoothness constraint and the OFC is modeled into an energy minimization framework. By calculus of variation, the associated Euler-Lagrange equation pair implies that each velocity component should satisfy the OFC and Laplace's equation, balanced by a Lagrange multiplier. Gauss-Seidel iteration converges to the solution. This global method generates a dense flow field. However, it suffers from an oversmoothing effect at the motion boundary. To overcome this problem, many anisotropic smoothness constraints have been proposed to replace Horn-Schunck's isotropic one. The associated anisotropy terms are generally intensity based (linear term, e.g., [10] by Nagel) or flow based (non-linear term, e.g., [3], [16]). Such constraints prohibit smoothing over intensity or motion boundaries. The extension from a spatial smoothness constraint to a spatio-temporal constraint has also been discussed in [3] and [16], and significant improvement has been reported. However, this non-linear spatio-temporal iteration process is computationally expensive.

In contrast to the global iterative scheme, Lucas-Kanade's constant flow [8] is local and therefore more robust to noise. Specifically, this method assumes a piecewise constant flow field. Thus the under-determined brightness constraint equations can be composed to form an overdetermined system. As long as the system has rank 2 (i.e., the region is not featureless, or all points are fully aligned), a least squares solution can be found straightforwardly. This constant flow constraint was then developed into a constant *vector* constraint, for example, the vector of flow and its first-order spatial derivatives in [13] by Shi-Tomasi and [4] by Campani-Verri. The advantage of using an overconstrained system is the robustness of the least squares solution to noise. However, it is also sensitive to discontinuities

0-7695-3067-2/07 $25.00 © 2007 IEEE
DOI 10.1109/DICTA.2007.42

of depth, intensity, motion and occlusion. For better performance, Lucas-Kanade suggested using a weighted overconstrained system. However, as the weights are given by the second order derivatives, and the possible outliers are still in the system, this only partly alleviates problems caused by intensity discontinuity. Another problem is the dramatically increasing size of the system with the increasing dimension of the constraint vector, to obtain a robust and stable solution. For example, in Campani-Verri [4], 41×41 equations are used for 6 unknowns.

This paper aims at dense, accurate flow computation with well preserved motion discontinuity. To this end, we construct a dynamic overconstrained system of equations to recover the optical flow, deformation and acceleration. An initial system of equations for flow estimation at each pixel is built up from selected pixels in its spatial neighborhood. Pixels are selected by detecting and rejecting outliers based on image intensity, first and second order spatial variation. Flow vectors obtained by solving this system are then tested using the brightness constraint equation. We use the residual of this equation as an indicator for the selection of temporal neighbors. Next, we use the selected temporal neighbors to refine the vectors obtained in the previous step. In the final step, we utilize the acceleration vectors to detect regions with possible occlusion or disocclusion. The velocity vectors in such regions are then corrected using flow from earlier or later frames, depending on whether occlusion or disocclusion is detected.

2. Problem Statement and Related Work

We build our system on the optical flow constraint, which assumes that the brightness E of a spatio-temporal scene point (x, y, t) remains constant over time. This assumption means that motion is the sole cause for the intensity change of a pixel. Mathematically, this constraint is expressed by the Optical Flow Constraint Equation (OFCE),

$$\frac{dE(x,y,t)}{dt} = E_x x_t + E_y y_t + E_t = 0. \qquad (1)$$

where, as throughout the paper, the subscripts denote the corresponding partial derivatives. Eq.(1) links the image velocity $(x_t, y_t)^T$ (or optical flow) to the image gradient $(E_x, E_y, E_t)^T$, modeling optical flow recovery as an inverse problem. To estimate the two unknown velocity components, at least one other constraint is needed. Lucas-Kanade's constant flow method solves the problem by an overdetermined system,

$$\begin{bmatrix} E_x(p_1) & E_y(p_1) \\ E_x(p_2) & E_y(p_2) \\ \vdots & \vdots \\ E_x(p_{N^2}) & E_y(p_{N^2}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -E_t(p_1) \\ -E_t(p_2) \\ \vdots \\ -E_t(p_{N^2}) \end{bmatrix}$$

where $p_1, p_2, \cdots, p_{N^2}$ are pixels within an $N \times N$ window, and (u, v) is the assumed constant flow of this window. The least squares solution can be obtained by computing the generalized inverse of the $N^2 \times 2$ matrix.

The work presented here is inspired by this strategy. By taking partial derivatives at both sides of Eq.(1) along the spatio-temporal axes, we obtain a linear system of 3 equations of 8 unknowns,

$$\begin{bmatrix} E_{xx} & E_{xy} & E_x & E_y & 0 & 0 & 0 & 0 \\ E_{xy} & E_{yy} & 0 & 0 & E_x & E_y & 0 & 0 \\ E_{xt} & E_{yt} & 0 & 0 & 0 & 0 & E_x & E_y \end{bmatrix} \vec{V} = \begin{bmatrix} E_{xt} \\ E_{yt} \\ E_{tt} \end{bmatrix}. \qquad (2)$$

where $\vec{V} = [u,\ v,\ u_x,\ u_y,\ v_x,\ v_y,\ u_t,\ v_t]^T$. The center four entries of $\vec{V}$ describe the dilation and vorticity of the moving object, and have been of particular research interest for time-to-contact and obstacle avoidance (e.g., [5]). The last two entries of $\vec{V}$ compose the acceleration vector, which will be used to detect occlusion/disocclusion in our method.

In most circumstances, it is adequate to approximate a small surface patch by a moving plane. Hence the velocity can be approximated by a quadratic polynomial in the image coordinates. Therefore, it is reasonable to assume that, within a small $N \times N$ image region, the vector $\vec{V}$ is nearly constant. This assumption leads to a $3N^2 \times 8$ overdetermined system of 8 unknowns, which we write as

$$A\vec{V} = b. \qquad (3)$$

The least squares solution of an overdetermined system is robust to noise, but sensitive to outliers. In optical flow computation, the outliers are pixels at an object boundary, a motion boundary, or an occlusion boundary. In these regions, either the brightness or the velocity (or both) is not constant.

The aim of this paper is to devise a system that avoids regularizing the irregular pixels. The irregularity is first detected from the image intensity and derivatives, and then the residual from the optical flow constraint, and finally from the acceleration.

Overdetermined systems based on the assumption of a constant vector consisting of flow and its first order spatial derivatives have been proposed previously in [4, 14, 11]. Although all these works and the one presented here are based on some overdetermined system that has a similar form to (3), they are fundamentally different. The additional constraint of [14] is that $d\nabla E/dt = 0$, whereas [4] assumes affine motion. Neither of these assumptions is employed here. [11] is a general discussion based on a Taylor expansion of intensity derivatives, investigating optimal combination schemes of the different-order differential equations

derived from OFCE. In contrast, our aim is to improve the estimation of flow at a point by dynamically forming a system of equations based only on pixels that are determined to be consistent with that point.

Although a model of occlusion has previously been incorporated into optical flow estimation, for example in [2], it has not been related to the constant vector constraint.

3. Dense Flow Computation with High Accuracy

3.1. Selection of spatial neighbors

Within the scope of this section, we assume that the optical flow constraint holds at the pixel of interest. The case where this constraint fails will be discussed in Section 3.3. In the following and throughout the paper, we denote the pixel of interest by c, the linear size of the surrounding searching window by N, and any other pixels in the window by p.

Our first selection criterion is pixel intensity similarity. Significant intensity difference between adjacent pixels generally implies object boundaries, and hence the high likelihood of inconsistent flow fields, if the objects move independently. Therefore, among all the candidates in the search region, we select only those with similar intensity values . Specifically, we base our selection on computing the intensity difference between c and the candidate pixel p_i, i.e.,$d_i = |E(p_i) - E(c)|, i = 1, 2, \cdots, N^2$. We then sort the set of d_i, i.e., $S = \left\{ d_i | \, i = 1, 2, \cdots, N^2 \right\}$, into set $\tilde{S} = \left\{ \tilde{d}_j \middle| \, j \in [1, N^2], \tilde{d}_j \in S, \tilde{d}_j <= \tilde{d}_{j+1} \right\}$ in ascending order. At most, the first $M(< N^2)$ pixels from the ordered set $\tilde{S}$ will be selected. Furthermore, if P_{j_0} is the first pixel such that $\tilde{d}_{j_0}$ is larger than a similarity threshold, then further pixels corresponding to $j \geq j_0$ are rejected. In our experiments on RGB sequences with 8 bit intensity levels for each channel, the intensity difference is computed as the sum of the absolute difference of the three channels. The similarity threshold $T = 96$, which means that the average absolute difference of the three channels should not exceed $1/8$ of the 256 intensity levels.

From the pixels selected in the previous step, we continue to detect and reject outliers with inconsistent gradients. The spatio-temporal gradient (E_x, E_y, E_t) is the direction of maximum intensity variation. If the image window corresponds to a small surface patch with smooth variation, pixels within the window are expected to have similar intensity gradient vectors. Thus inconsistent gradient vectors generally indicate regions containing an object boundary or depth discontinuity. Therefore in this step we reject pixel p if

$$\begin{cases} sign(E_x/E_t)(p) = -sign(E_x/E_t)(c) \\ sign(E_y/E_t)(p) = -sign(E_y/E_t)(c) \end{cases}$$

or

$$\begin{cases} sign(E_{xt}/E_{tt})(p) = -sign(E_{xt}/E_{tt})(c) \\ sign(E_{yt}/E_{tt})(p) = -sign(E_{yt}/E_{tt})(c). \end{cases}$$

The last step in spatial neighbor selection is to reject outliers caused by occlusion and disocclusion. Here, by disocclusion, we mean regions that have been occluded and then reappear in the sequence. If a pixel is to be occluded/disoccluded, the constant brightness assumption is invalid. As a consequence, the differential constraint equations derived from it will skew system (3). We reject such outliers based on the observation that the temporal derivatives of these tend to have large magnitude compared to their spatial derivatives. Therefore, if a pixel p meets any of the following inequalities, it is rejected:

$$\frac{|E_t|}{|E_x| + |E_y|} > T_1,$$

$$\frac{|E_{xt}|}{|E_{xx}|} > T_2,$$

$$\frac{|E_{yt}|}{|E_{yy}|} > T_3,$$

where the thresholds T_1,T_2 and T_3 are the corresponding mean values over the whole image.

It is possible that the selection procedure results in a matrix A with rank ≤ 8. In this case, we choose 6 pixels that have most similar intensity values to form system (3). Now the initial dense flow field can be obtained from $\vec{V} = A^{+}b$. The flow vectors with low fidelity will be refined or corrected in the following sections.

3.2. Selection of temporal neighbors

In general, the perceived object velocity is closely correlated between successive frames. Therefore, the spatial smoothness constraint can be extended to a spatial-temporal one. On one hand, integrating temporal neighbors into system (3) improves the robustness of the recovery to noise and time aliasing. On the other hand, this extension oversmoothes motion boundaries if the region contains independently moving objects. Therefore the extension from spatial constraint to spatio-temporal constraint must be implemented with care. Weickert-Schnorr's total variation method uses anisotropy terms to steer the energy minimization formulation, by assigning higher weights to spatio-temporal neighbors of similar intensity or motion pattern. They report a significant improvement in average

angular error due to this extension. So far in the literature, the spatio-temporal version of Lucas-Kanade's or similar approaches is simply to extend the constant flow (and its derivatives) constraint to a spatio-temporal $N \times N \times N$ neighborhood.

In this section, we propose selecting the coherent temporal neighbors based on the vector obtained in Section 3.1. To do so, we employ the residual error given by

$$r = |E_x u + E_y v + E_t| . \tag{4}$$

This residual generally measures the fidelity of the flow obtained in the previous step. Ideally, if the recovered flow is the ground truth, r is expected to have a small value, except when OFCE itself is invalid, which will be discussed later. In particular, small r means that the constraints in previous sections are appropriate, and it is then safe to apply them to temporal neighbor selection. By contrast large r implies the presence of discontinuity factors in the neighborhood. Thus we only extend the constant vector constraint to temporal neighbors if r is smaller than a threshold.

The selected temporal neighbors form a new system

$$R\vec{V} = d,$$

which is combined with the previous system (3) to form

$$W\vec{V} = \begin{bmatrix} A \\ R \end{bmatrix} \vec{V} = \begin{bmatrix} b \\ d \end{bmatrix}. \tag{5}$$

Directly computing $W^+ = (W^T W)^{-1} W^T$ for the least squares solution of (5) is computationally inefficient. Instead, we compute W^+ from R and A^+, which we have obtained from previous steps, by the following scheme,

$$Q = \left(RA^+\right)^T;$$

$$B_1 = (I + QQ^T)^{-1}(A^+)^T;$$

$$B_2 = Q^T B_1;$$

$$W^+ = [B_1|B_2].$$

The interested reader is referred to [12] for the mathematical proof that $[B_1|B_2]$ is the generalized inverse of W.

3.3. Correction of Erroneous Flow

The formulation in previous sections is based on the validity of the constant brightness constraint. However, when the scene contains intensity discontinuity or occlusion/disocclusion, the image brightness is non-differentiable, and hence even the true flow may not satisfy the OFCE. If the invalidity of the constraint is caused by fast intensity variation, system (5) can still estimate the flow accurately, given the motion is smooth in a sufficiently large region. This is because system (5) treats sporadic fast variation as noise, and the main contribution to the system is from the differentiable neighbors. The problem is more complicated if the validity is due to occlusion/disocclusion or motion discontinuity, because most pixels in the system have inconsistent motion patterns.

Such regions can be detected from the computed acceleration components u_t and v_t, which tend to have large magnitude if system (5) contains pixels with inconsistent motion. We observe that $|u_t| + |v_t|$ is an efficient detector for pixels with erroneously computed flow. This measure has low computational cost and signals disoccluded/occluded regions well. Once such pixels are detected, the velocity vectors in the surrounding regions (i.e. the search window) are examined for occlusion/disocclussion. More specifically, we divide the window into four regions: the left half Ω^l, the right half Ω^r, the upper half Λ^u and the lower half Λ^l. The average velocity components are calculated in these rectangular sub-regions by

$$\begin{aligned} v_0^l &= mean(v(p)), \quad p \in \Omega^l; \\ v_0^r &= mean(v(p)), \quad p \in \Omega^r; \\ u_0^l &= mean(u(p)), \quad p \in \Lambda^l; \\ u_0^u &= mean(u(p)), \quad p \in \Lambda^u; \end{aligned}$$

Here, without loss of generality, we assume v is the velocity's horizontal component, and u the vertical component. The signed magnitude and convergence/divergence of these average velocity components are used to judge the occurrence of occlusion/disocclusion. For example, convergence of vectors $(0, v_0^l)$ and $(0, v_0^r)$ indicates that regions Ω^l and Ω^r are tending to merge, and therefore that the size of the occluding region is increasing over time. Divergence of the two vectors indicates that the two regions are moving apart, and hence disocclusion, or decrease of the occluding region, is inferred. If $(0, v_0^l)$ and $(0, v_0^r)$ point to the same direction, for example to the left, then faster/slower v_0^r indicates occlusion/disocclusion, compared to v_0^l. A similar argument holds for the velocity component u.

Regions with occlusion and disocclusion are treated differently. To correct the obtained flow in a region of increasing occlusion, we resort to the preceding frames to form a new system (5). Conversely, for the disocclusion region, subsequent frames are used since they are less likely to include the motion boundary.

4. Experimental Results

In this section, we present our experiments conducted on a color sequence *Street* and an 8-bit grayscale sequence *Hamburg Taxi*. Sequence *Street* was created to evaluate optical flow techniques in [9] by McCane-Novin-

Crannitch-Galvin and [6] by Galvin et al. The sequence and the ground-truth flow are publicly available at www.otago.ac.nz/research/vision. Figure 1 shows its 10th frame and Figure 2 is the zoomed-in view of the true flow around the car region, subsampled by a factor of 2. This sequence depicts a street scene with a car moving to the right and the camera moving to the left. Sequence *Street* is challenging for optical flow recovery, as it contains a variety of factors that violate the optical flow constraint. These factors include shadows, transparency, intensity discontinuity, depth discontinuity, different motion patterns, and occlusion/disocclusion.

In our experiment, the sequence is pre-smoothed by a Gaussian kernel along all three dimensions before calculating the derivatives. The standard deviation of the smoothing kernel is 0.8 pixel, and the kernel length is 7 pixels. The first order derivative is approximated by the 3-point central differencing filter $\frac{1}{2}[-1\ 0\ 1]$. The second order derivatives are obtained by cascading the first-order derivatives. The size of the searching window is 19×19 pixels, with at most 225 pixels selected from each window for the overdetermined system in the first step. The second step of the selection process is conducted on the two adjacent frames. If occlusion/disocclusion is detected, the flow is corrected using the preceding/subsequent second and third adjacent frames.

Figure 3 demonstrates the flow recovered by our method, zoomed-in and subsampled by a factor of 2. This figure shows that the motion boundary is well preserved. To quantitatively evaluate the accuracy of our method, we measure the recovered flow by the average and standard deviation of the angular error, as suggested by Barron-Fleet-Beauchemin in [1] . These quantitative measurements are widely used in the current literature of optical flow computation. The angular error is defined by

$$arccos\left(\frac{u_c u_e + v_c v_e + 1}{\sqrt{(u_c^2 + v_c^2 + 1)(u_e^2 + v_e^2 + 1)}}\right),$$

where $(u_c, v_c, 1)$ is the true velocity vector, and $(u_e, v_e, 1)$ is the estimated one.

We compare our methods to three existing differential techniques in the literature: Lucas-Kanda [8], Uras-Girosi-Verri-Torre [15] and Weickert-Schnörr [16]. Methods of Lucas-Kanade and Uras et. al are reported as two of the best differential techniques in [1] by Barron-Fleet-Beauchemin (1994); and Weickert-Schnörr's method has achieved the best performance on *Street* in the literature by far. All methods are compared over full density flow field. Our experimental settings of the first two are as same as the ones employed in [1]. Table 1 lists the average angular error and standard deviation achieved by each method. The numerical comparison shows that the proposed method outperforms existing techniques, and the associated flow field (Figure 3, with ground truth in Figure 2) shows a particular improvement in areas near occlusion boundaries.

The algorithm is also tested on the real image sequence *Hamburg Taxi*, as in [1] and [16]. Figure 4 shows a sample frame of the sequence. The scene contains four moving objects: a turning taxi, a car, a van and a pedestrian. Challenging factors of this sequence include its low resolution and occlusion. For example, the tree in the lower right occludes parts of the van, when the van is moving from left to right. In our experiment on this sequence, no pre-smoothing is implemented, as suggested in [16]. The other settings are the same as those of the experiment on sequence *Street*. Figure 5.a and Figure 6.a are the computed flow fields around the car and the van respectively, before occlusion/disocclusion detection and flow correction. Figure 5.b and Figure 6.b present the finally computed flow fields, which show the improved performance at motion-boundaries. Note that the tree trunk in the lower right of Figure 6, which is still and occludes the van, is successfully partitioned from the moving van. Also note that the moving shadow in front of the car in Figure 5 is detected more accurately as not moving after occlusion/disocclusion detection.

	Average Error	Standard Deviation
Uras et al.	13.27	19.68
Lucas-Kanade	8.47	13.03
Weickert-Schnörr	4.85	unspecified
The proposed method	3.95	9.77

Table 1. Comparison with existing differential techniques of optical flow computation

5. Conclusion

This paper has demonstrated that the accuracy of dense optical flow estimation can be improved by selective use of pixels in a point's spatial and temporal neighbourhood. In particular, the estimation of flow at occlusion boundaries is enhanced by distinguishing between points that have been occluded in the recent past and those that will be occluded in the near future. The computational cost of including these extra checks is minimal.

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):231–245, 2000.

Figure 1. The 10th frame of the test sequence *Street*.

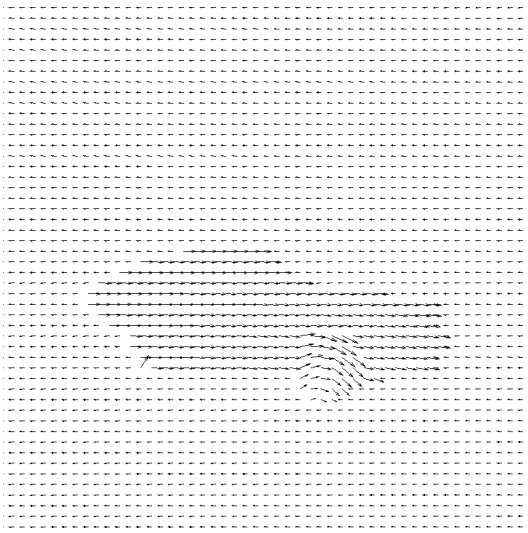

Figure 2. The true optical flow vectors, subsampled by a factor of 2.

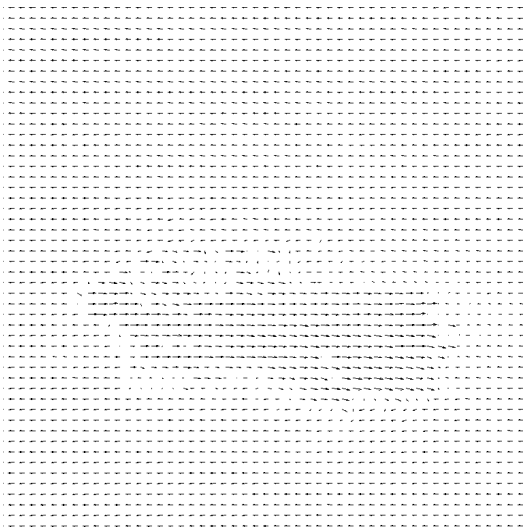

Figure 3. The flow vectors computed by the proposed method, subsampled by a factor of 2. Flow is computed more accurately near the front of the car than for other methods.

Figure 4. The 30th frame of the test sequence *Hamburg Taxi*.

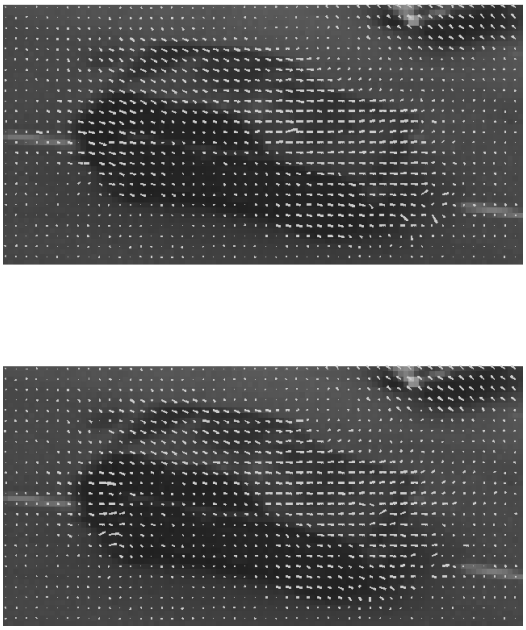

Figure 5. (a) Top: the optical flow computed around the car before occlusion/disocclusion detection and erroneous flow correction; (b) Bottom: the optical flow computed after occlusion/disocclusion detection and erroneous flow correction. The flow field in the shadowed region in front of the car is detected more accurately.

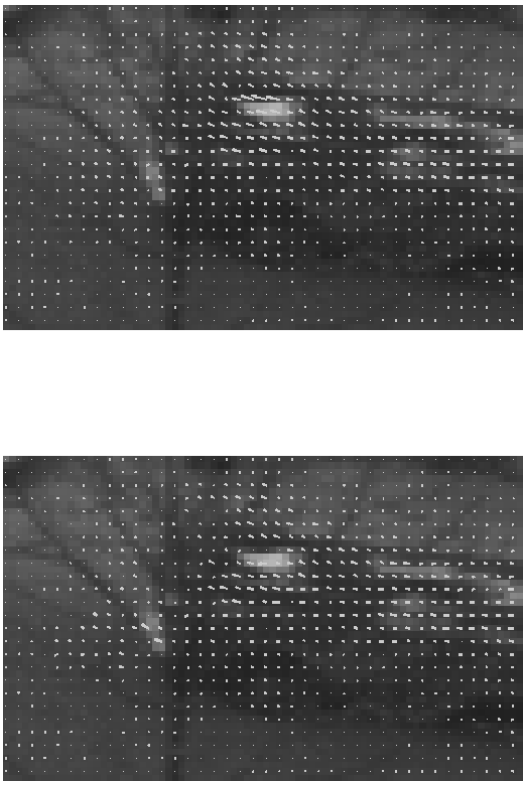

Figure 6. (a) Top: the optical flow computed around the van before occlusion/disocclusion detection and erroneous flow correction; (b) Bottom: the optical flow computed after occlusion/disocclusion detection and erroneous flow correction. The tree is more accurately segmented from the passing van.

[3] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[4] M. Campani and A. Verri. Computing optical flow from an overconstrained system of linear algebraic equations. *IEEE Conference on Computer Vision*, pages 22–26, December 1990.

[5] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. *Lecture Notes in Computer Science*, 588:187–202, 1992.

[6] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An analysis of eight optical flow algorithms. *Proc. of British Machine Vision Conference*, September 1998.

[7] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[9] B. McCane, K. Novins, D. Crannitch, and B. Galvin. On benchmarking optical flow. *Computer Vision and Image Understanding*, 84(1):126–143, 2001.

[10] H.-H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision Graphics and Image Processing*, 21(1):85–117, January 1983.

[11] M. Otte and H.-H. Nagel. Optical flow estimation: Advances and comparisons. *ECCV 94, Lecture Notes in Computer Science*, 800:51–60, 1994.

[12] J. V. V. Rao. Some more representations for the generalized inverse if a partitioned matrix. *SIAM Journal of Applied Mathematics*, 24:272–276, 1973.

[13] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, June 1994.

[14] M. Tistarelli. Multiple constraints to compute optical flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(12):1243–1250, December 1996.

[15] S. Uras, F. Girosi, A. Verri, and V. Torre. A computational approach to motion perception. *Biological Cybernetics*, 60:79–97, 1988.

[16] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14:245–255, 2001.