Estimating 3D Flow for Driver Assistance Applications

Jorge A. Sánchez [1], Reinhard Klette [2], and Eduardo Destefanis [1]

[1] Universidad Tecnológica Nacional, Facultad Regional Córdoba
Cordoba, Argentina

[2] The *.enpeda..* Project, The University of Auckland
Auckland, New Zealand

Abstract. This paper proposes a technique for estimating 3D flow vectors, by combining a KLT tracker with subsequent scale-space analysis of tracked points. A tracked point defines a 2D vector, which is mapped into 3D space based on ratios of maxima of scale-space characteristics. The approach is tested for night-vision sequences as recorded (at Daimler AG, Germany) for driver assistance projects. Those image sequences (at 25Hz) are characterized by being slightly blurry and of low contrast.

Key words: motion analysis, motion vector fields, 3D motion, driver assistance

1 Introduction

The estimation of dense motion fields is still a challenging task for vision-based driver assistance systems (DAS), where motion vectors are often relatively long even if sequences are taken at a frame rate of more than 30 Hz. This paper suggests a way to derive 3D directions of observed 2D motion vectors, which allows a more consistent interpretation of motion fields.

Note that a 3D direction of a motion vector is not yet defining its pose, which would also require to identify its position (e.g., via stereo analysis). The 3D pose of projected motion vectors is known as *scene flow*. Scene flow techniques crucially depend on whether a sparse or a dense representation is desired, or whether motion is assumed to be rigid or not.

Sparse representations involve some kind of spatio-temporal feature matching; for the monocular case this is accomplished by methods known from structure-from-motion (SfM), which usually assumes a rigid motion of the whole scene [13]. If there is more than one view available, as in binocular stereo, the computation of scene flow relies on integration of depth and temporal information in some cooperative way [16]. For the case of dense representations, this involves the minimization of energies in a variational framework in order to add some smoothness constraint, needed to deal with the aperture problem [15].

Our approach tries to use information provided by observed temporal changes in size (scale) of local image regions if a single camera moves relatively to a scene. (This is known to be a very important source of information for the visual perception of motion.) We use a scale-space representation of consecutive image frames in order to obtain (for each tracked point) a measure for the diameter of image brightness patterns (that surround tracked points), as established by [4–6] for automatic scale selection.

The idea is to identify a *characteristic scale* to be the value where a normalized differential entity takes a local extrema. Such scale values are measured in terms of standard deviations of the Gaussian kernel which are used to generate corresponding levels of the scale-space representation.

In [11] it is experimentally shown that the Laplacian of Gaussian (LoG) is the most stable in a considered set of differential normalized operators, possibly used for scale selection. Thus, this operator is selected for the scale selection stage described in Section 4. The remainder of the paper is organized as follows: Section 2 presents equations for analyzing directions of motion; Section 3 describes the used tracking scheme; the complete algorithm is presented in Section 5; finally, Sections 6 and 7 present experimental results and conclusions.

2 Estimation of 3D Directions

We consider a 3D point P, tracked between frames I_t and I_{t+1}, and propose a possible way for calculating the 3D direction of the observed motion.

2.1 Update Equation

Consider a disk of radius ρ moving towards an ideal pinhole-type camera of focal length f. Without loss of generality, let the radius move parallel to the Y-axis of the XYZ-camera coordinate system (i.e., $r = Y_c - Y_e$, for center P_c and an edge point P_e of the disk). A 3D point $P = (X, Y, Z)$ in the world (in camera coordinates) projects into a point $p = (x, y, f)$ in the image plane, with $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$. Point P_c projects into $p_c = (x_c, y_c, f)$, and P_e projects into $p_e = (x_e, y_e, f)$. The moving disk is at time t at distance Z_t, and projected

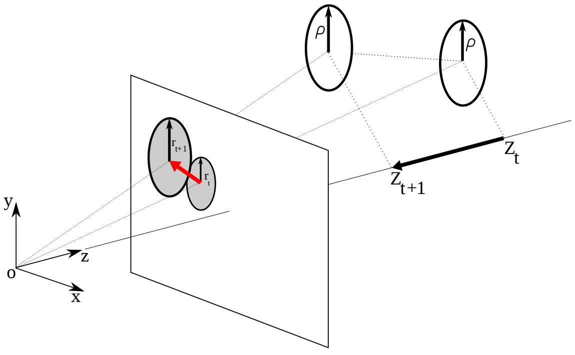


Fig. 1. Two projections of a moving disk, at times t and $t + 1$.

into image I_t as a disk of radius r_t. We obtain the following for the area of this projected disk:

$$A_t = \pi r_t^2 = \pi \left(y_c - y_e\right)^2 = f \frac{\pi}{Z_t^2} \left(Y_c - Y_e\right)^2 = \pi f \frac{\rho^2}{Z^2}$$

Radius ρ of the disk is constant over time, thus, the product $A_t Z_t^2 \sim \rho^2$ will also not change over time.

We consider projections of the disk at times t and $t+1$. Because the ratio of square roots of areas is proportional to the inverse of the ratio of corresponding Z-coordinates of the disk, we are able to define a *z-ratio*

$$\mu_z = \frac{\sqrt{A_t}}{\sqrt{A_{t+1}}} = \frac{Z_{t+1}}{Z_t} \tag{1}$$

either by area or Z-values. Such a z-ratio can also be defined just for a pair of projected points $P_t = (X_t, Y_t, X_t)$ and $P_{t+1} = (X_{t+1}, Y_{t+1}, Z_{t+1})$ (just by the ratio of Z-coordinates).

Using the central projection equations for both projected points, we obtain for their *x-ratio* and *y-ratio* the following:

$$\mu_x = \frac{X_{t+1}}{X_t} = \frac{Z_{t+1}}{Z_t} \cdot \frac{x_{t+1}}{x_t} = \mu_z \frac{x_{t+1}}{x_t} \tag{2}$$

$$\mu_y = \frac{Y_{t+1}}{Y_t} = \frac{Z_{t+1}}{Z_t} \cdot \frac{y_{t+1}}{y_t} = \mu_z \frac{y_{t+1}}{y_t} \tag{3}$$

Altogether, this may also be expressed by the following *update equation*:

$$\begin{pmatrix} X_{t+1} \\ Y_{t+1} \\ Z_{t+1} \end{pmatrix} = \begin{pmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & \mu_z \end{pmatrix} \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} \tag{4}$$

with μ_x, μ_y, and μ_z as in Equations (2), (3), and (1), respectively. In other words, knowing μ_z and ratios $\frac{x_{t+1}}{x_t}$ and $\frac{y_{t+1}}{y_t}$ allows to update the position of point P_t into P_{t+1}. Assuming that P_t and P_{t+1} are positions of one tracked 3D point P, from time t to time $t+1$, we only have to solve two tasks: (1) decide for a technique to track points from t to $t+1$, and (2) estimate μ_z. If an initial position P_0 of a tracked point P is known then we may identify its 3D position at subsequent time slots. Without having an initial position, we only have a 3D direction P_t to P_{t+1}, but not its 3D position.

2.2 3D Direction of Projected Motion

Consider a mobile platform moving on a planar surface, as illustrated in Figure 2. The relative motion of a point in 3D space can be expressed (with respect to the camera coordinate system) by the following increments:

$$\begin{aligned} \mathit{\Delta} X &= X_{t+1} - X_t = (\mu_x - 1) X_t \\ \mathit{\Delta} Y &= Y_{t+1} - Y_t = (\mu_y - 1) Y_t \\ \mathit{\Delta} Z &= Z_{t+1} - Z_t = (\mu_z - 1) Z_t \end{aligned}$$

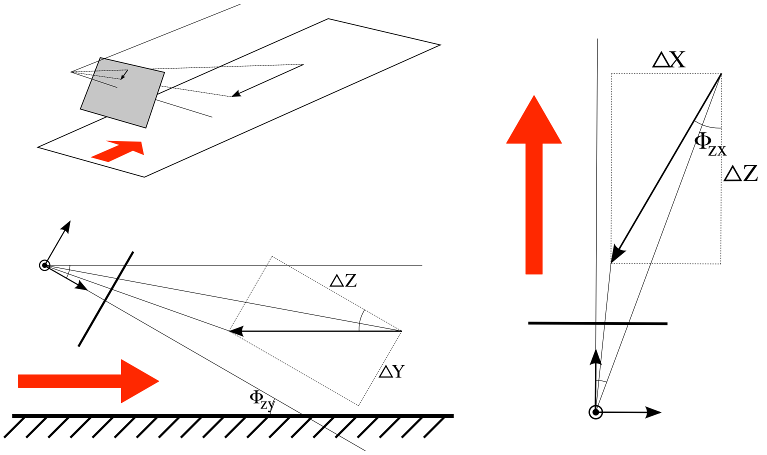


Fig. 2. A tilted camera translating along a plane (top left), motion angles on the ZY-plane (bottom left), and on the ZX-plane (right).

The ratios

$$\frac{\Delta X}{\Delta Z} = \left(\frac{\mu_x - 1}{\mu_z - 1}\right)\frac{X_t}{Z_t} = \left(\frac{\mu_x - 1}{\mu_z - 1}\right)\frac{x_t}{f} \tag{5}$$

$$\frac{\Delta Y}{\Delta Z} = \left(\frac{\mu_y - 1}{\mu_z - 1}\right)\frac{Y_t}{Z_t} = \left(\frac{\mu_y - 1}{\mu_z - 1}\right)\frac{y_t}{f} \tag{6}$$

of those increments are the tangents of the *navigation angles* Φ_{zx} and Φ_{zy} (see Figure 2), respectively, that represent the *3D direction of motion* (between two subsequent frames) for a tracked 3D point.

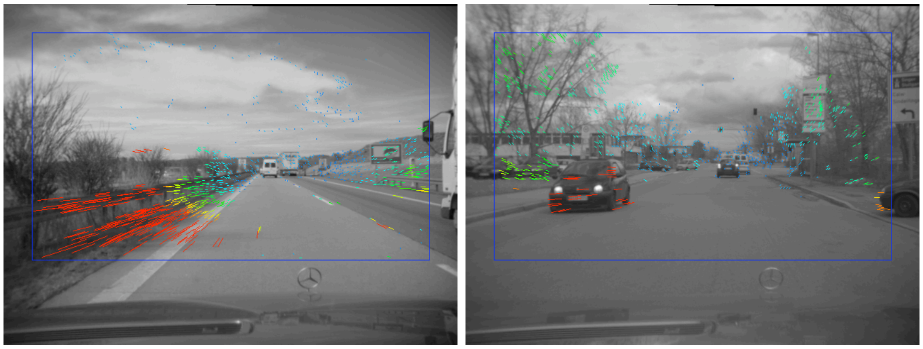


Fig. 3. Optical flow computed for Sequences 1 and 2.

3 Feature Tracking and Test Data

The determination of those navigation angles relays on the detected projected motion of tracked points in the image plane. Tracking methods for estimating a dense 2D motion field are known as *optical flow techniques*.

Actually, the used tracking method is not essential for presenting the basic idea of our approach for estimating 3D directions; however, it is, of course, important for obtaining reliable results of the proposed approach.

For this paper we simply used an open implementation [3] of the Lucas-Kanade [9] feature tracker, with initial points selected as in [12].

Regarding test sequences, we decided for Set 1 (seven night vision stereo sequences, provided by Daimler AG) as available on [2]. Those will be called 'Sequence 1', 'Sequence 2', and so forth, as on this website and described in [7]. Additionally, also one 'Desktop sequence' was generated and used for performance analysis. This desktop sequence was generated by translating a calibrated camera along an optical bench, with constant 3D viewing angles relatively to the surface of the desktop.

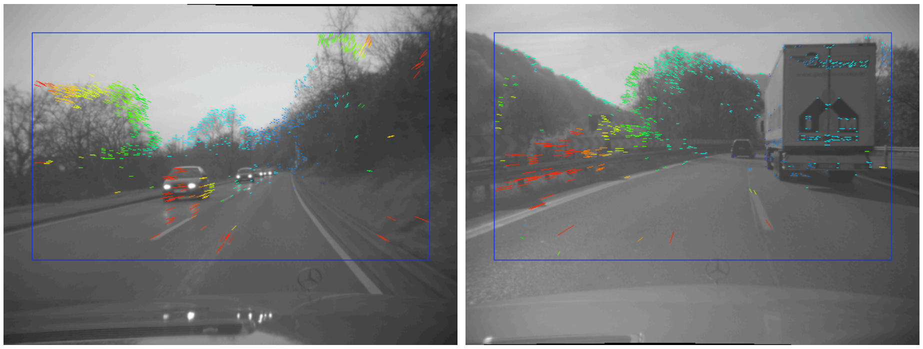


Fig. 4. Optical flow computed for Sequence 3 and 4.

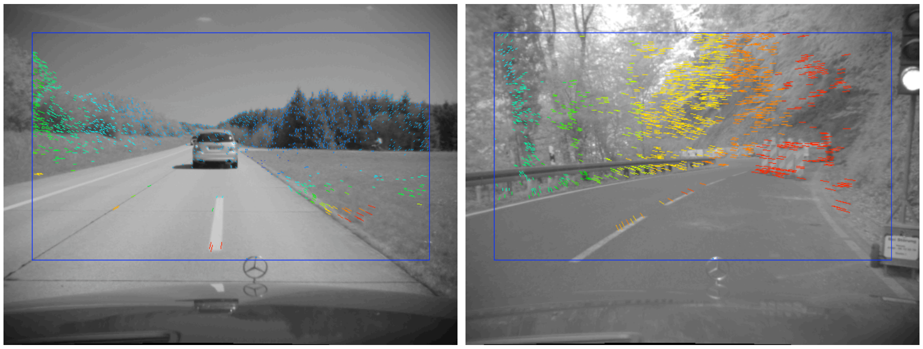


Fig. 5. Optical flow computed for Sequence 5 and 6.

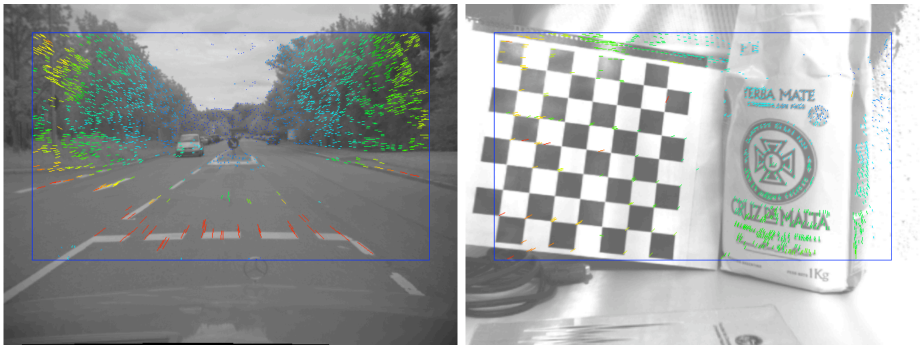


Fig. 6. Optical flow computed for Sequence 7 and the Desktop sequence.

Figures 3 to 6 illustrate tracking results for Sequences 1 to 7, and the Desktop sequence. By using relatively high thresholds, only relatively sparse motion fields are shown. The used coloring is based on the length of the optic flow vectors.

4 Scale Ratio for a Tracked Point

The idea behind the presented approach is as follows: instead of directly tracking image regions (such as disks, as discussed in Section 2), single feature points are tracked, but an 'area of influence' is assigned to such a point, basically taking the role of a tracked disk.

For tracked points, a measure is computed for the 'extension of the local image structure' in a local (or semi-local) neighborhood. Such measures, computed independently for each pair of points (i.e., a 2D flow vector between time t and $t+1$), are then used to determine a scale ratio of associated intensity profiles 'surrounding' those feature points, which is finally used as an estimate of the z-ratio μ_z.

The approach for detecting scale-ratios follows *scale space theory* as discussed in [5, 6]. Note that this is only one option; similar to the variability when deciding for one optical flow technique, also an alternative method may be used for scale-ratio estimation.

We briefly recall scale space theory. Given an image function $I : \mathbb{R}^2 \rightarrow \mathbb{R}$, their scale space representation $L : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$ can be obtained by convolutions

$$L(p, \sigma) = (g_\sigma * I)(p)$$

of image I with a Gauss kernel g_σ, obtained by the Gauss function $G_\sigma : \mathbb{R}^2 \rightarrow \mathbb{R}$,

$$G_\sigma(p) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2} p^T p}$$

parameterized by standard deviation $\sigma \geq 1$.

In [6], a method for automatic scale selection is proposed, based on the evolution over scales of (possibly non-linear) combinations of normalized derivatives of $L(p, \sigma)$. The scale level at which such a response takes a local maxima is assumed to reflect the *characteristic diameter* of the surrounding data. The operator used in our experiments for scale selection is the normalized Laplacian, which is defined by

$$\nabla_{norm} L(p, t) = \sigma^2 \left| (D_x^2 L)(p, \sigma) + (D_y^2 L)(p, \sigma) \right| \tag{7}$$

where D_x^2 and D_y^2 are the second order derivatives of L (at scale level σ).

We use Figure 7 for visualizing the scale selection principle. In this simple example, the image on the left contains three white disks with center points p_1, p_2, and p_3. On the right, the figure shows the scale evolution of (7) for those three center points. In this example, the ratio between the scales, identified by maxima of the scale characteristics of those three center points, equals the ratio of areas of the corresponding white disks.

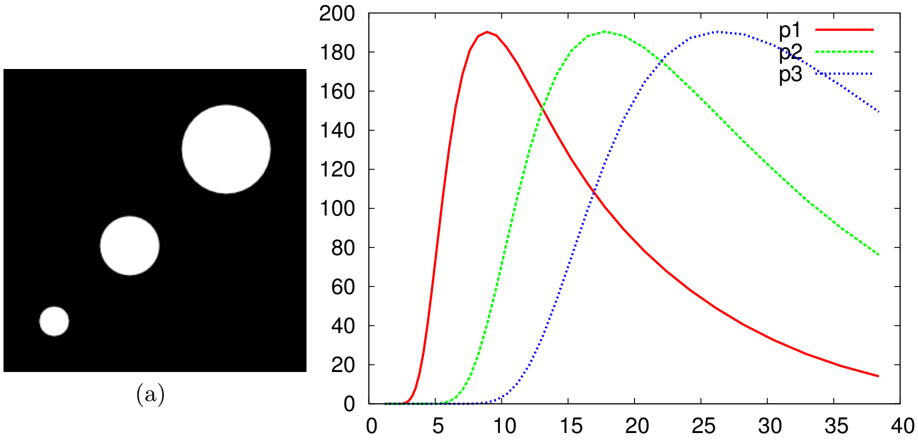
Given two consecutive frames I_t and I_{t+1} of a sequence. We calculate their scale space representations $L_t(p, \sigma)$ and $L_{t+1}(p, \sigma)$, for selected scales σ. For each selected pair of points (as a result of the tracking algorithm), we follow their scale characteristics for the normalized Laplacian of $L_t(p, \sigma)$ and $L_{t+1}(p, \sigma)$, and identify local maxima over the selected scales.

The function

$$c(\sigma) = K \sigma^p e^{-\sigma/\theta}$$

is used in order to obtain sub-scale estimates, where the parameters of this function are directly computed from the local maxima, extracted as an initial estimate, and its both neighboring values σ_1 and σ_2. Applying this approach, the maximum (i.e., magnitude)

$$K = c(\sigma_1) \sigma_1^{-a} e^{\sigma_{max}/\theta}$$



(a)

(b)

Fig. 7. (a) White disks of radius 25, 50 and 75 pixel. (b) Evolution on the Laplacian for center points p_i, for $i = 1, 2, 3$.

is identified at a sub-scale value $\sigma_{max} = a \cdot \theta$ The scale matching process between both projections p_t and p_{t+1} of the tracked point is then based on the magnitude K of the interpolation function at the scale level σ_{max}.

The determination of the scale ratio, associated to characteristic diameters of a tracked pair of points, allows to estimate the corresponding μ_z (see Equation (1)) and thus also to estimate the navigation angles (see Equations (5) and (6)).

After the estimation of the two navigation angles (for all pairs of tracked points), histograms of these two values are computed. This allows to calculate one *summarizing 3D direction*, for all the detected 3D directions between images I_t and I_{t+1}. This summarizing 3D direction results from

$$\hat{\phi}_{zx} = \arg\max_\theta h(\theta_{zx})$$
$$\hat{\phi}_{zy} = \arg\max_\theta h(\theta_{zy})$$

where $h(\theta_{zx})$ and $h(\theta_{zy})$ are the histograms of the navigation angles. Figure 8 shows two generated histograms.

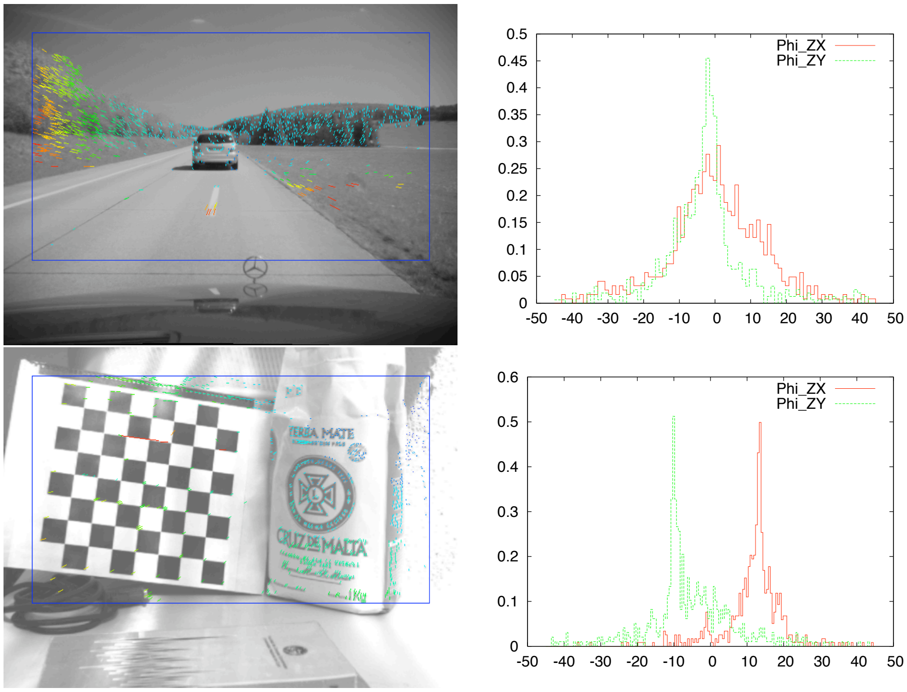


Fig. 8. Histogram examples for Sequence 5 and the Desktop sequence.

5 The Algorithm

To summarize, given a pair of consecutive images, I_t and I_{t+1}, the proposed algorithm consists of the following steps:

1. Compute the optical flow between images I_t and I_{t+1}, as described in Section 3.
2. For the same images, obtain their scale-space representation, $L_t(p, \sigma)$ and $L_{t+1}(p, \sigma)$, for a given set of predefined scales σ.
3. Select characteristic scales for the corresponding points resulting from Step 1.
4. Compute the scale-ratio from the detected scale maxima as $(\sigma_{t+1})_{max}/(\sigma_t)_{max}$ and compute the corresponding μ–factors.
5. Obtain the motion angles as the arcus tangent of (5) and (6).

For the construction of the scale space representations (Step 2), an exponential sampling was used, where the sigma value for scale level (n) is obtained from scale level $(n-1)$ as

$$\sigma_n = k\sigma_{n-1} = k^n\sigma_0 \quad \text{for} \;\; n = 1, 2, \ldots, N-1$$

This allows the use of the *Difference of Gaussians* (DoG) operator as a fast approximation technique of the Laplacian operator, as in [8]. Here, given σ_0 and σ_k, a scale space representation consisting of N levels is computed for I_t and I_{t+1}, respectively. In practice, 17 levels are computed with $\sigma_0 = 1$ and $k = 1.2$.

Given a scale selection operator, the scale selection for points in Step 3 consists of analyzing the evolution over scales of the operator's response (i.e., if we consider the scale-space representation as a three-dimensional space with coordinates (x, y, σ), the characteristic scale(s) for point (x_0, y_0) are those values of σ at which the magnitude of the response takes a (local) maxima). In the common

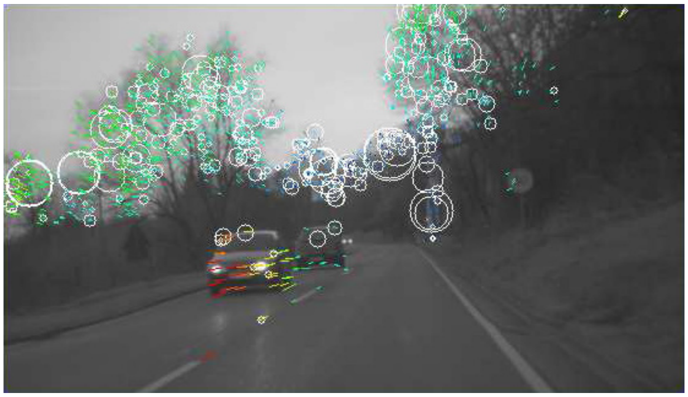


Fig. 9. Tracked points and detected scales (defining the size of the shown disks).

case in which more than one local maxima are detected, the magnitude of the response can be used as a measure of similarity.

Figure 9 shows an example of computed flow vectors and their corresponding characteristic scales (detected for those points for which a maxima over the scales of the Laplacian was correctly found); the size of disks represents the corresponding characteristic scale.

6 Results for Desktop and Daimler Sequences

Figure 10 shows some of the results obtained for Sequences 3 and 7, and the Desktop sequence.

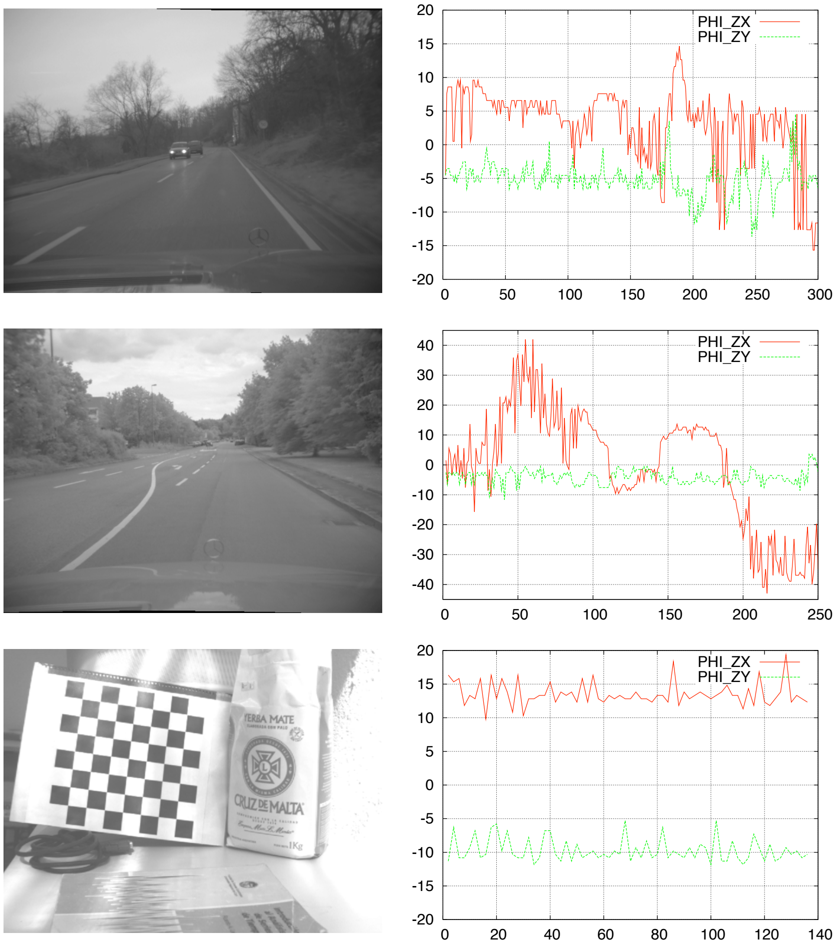


Fig. 10. Optical flow computed for Sequences 3, 7 and for the Desktop sequence.

The accuracy of the estimated values depends in some way on the magnitude of the relative motion. For points that remain almost static, and where the scale ratio y is close to 1, the quotients in Equations 5 and 6 are not well defined, due to noisy measurements. In our computations, points pairs with $|\sigma_{max}/\sigma_{min} - 1| < 10^{-6}$ are discarded, where σ_{max} (σ_{min}) denotes the largest (smallest) detected scale of tracked points. An upper limit to the μ_z's was also imposed, allowing only values < 5.

The Desktop sequence was generated with a calibrated camera, with translational motion on a rail with fixed navigation angles of approximately $\Phi_{zx} = 12°$ and $\Phi_{zy} = -10°$.

In the case of the Desktop sequence, mean and standard deviation of Φ_{zx} are equal to 13.531° and 2.93874°, respectively. For Φ_{zy}, those values are equal to $-10.3073°$ and 2.93874°, respectively, taken over the entire sequence. In the case of the Daimler sequences, there was no ground truth available (estimated), and they were only used as a qualitative (visual) reference. For example, for Sequence 7 (see example in Figure 10), estimated directions Φ_{zx} correspond 'quite well' to the steering of the car over the sequence. Estimated values Φ_{zy} remain at about 4°. In Sequence 2 we observed a low frequency oscillation in the value of Φ_{zy}, starting about at frame 180, when the 'squirrel' (actually, a cat) crossed the street and the car made a breaking maneuver.

In all cases, observed noise is mainly due to the scale matching subprocess. As given, for example, projective distortions of local image patches are not taken into account, causing possibly some serious underestimations of scale factors. The proposed (non-run-time-optimized) algorithm runs at approximately 1 fps on a 3.0 GHz Intel© Core 2 Duo CPU.

7 Conclusions

A method for the instantaneous (frame to frame) estimation of the 3D direction of motion was proposed and studied, based on the determination of scale ratios between tracked points. The critical issue is the accurate scale estimation step. The MSER region extractor [10] is an alternative (to the presented choice) option and possibly a more robust way for the determination of scales (characteristic diameters), where local image patches can be represented with affine invariance, serving as a first-order approximation for (more general) projective deformations induced by the relative motion of the camera.

Besides some poor estimations for some frames, the proposed method may be recommended as a possible approach for the use of perceptually very important spatio-temporal cues induced on images as an observer moves relatively to the scene. The extracted information has the advantage of being local, and allowing robustness in the case of multiple moving objects. The same principle of scale-ratio estimation could also be used for motion segmentation, or to add new constraints to multiple-view approaches of 3D motion estimation, thus further contributing to the already known coherence between optic flow vectors and image disparities.

The overall run-time of the algorithm can be significantly improved by the use of dedicated hardware (FPGA/ASICs). Here, the bottle-neck remains in the computation of the scale-space representations of the given images. This is done, as mentioned previously, by means of convolution with bi-dimensional Gaussians of variable width. This type of kernels allows for efficient implementations in terms of separable 1D Gaussians (which allow recursive implementations; see [1, 14]).

References

1. Deriche, R.: Recursively implementing the Gaussian and its derivatives In: Proc. *2nd Int. Conf. on Image Processing* (1992), 263–267
2. *.enpeda..* Image Sequence Analysis Test Site, `http://www.mi.auckland.ac.nz/6D/` (follow the *data* link)
3. Intel Open Source Computer Vision Library, `http://www.intel.com/research/mrl/research/opencv/`
4. Lindeberg, T.: On scale selection for differential operators, In: *ISRN KTH/NA/P–93/12–SE* (1993), 857–866
5. Lindeberg, T. : *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, Norwell, MA, USA (1994)
6. Lindeberg, T.: Feature detection with automatic scale selection, *Int. J. Computer Vision* **30** (1998) 77–116
7. Liu, Z., Klette, R.: Performance evaluation of stereo and motion analysis on rectified image sequences. Technical report, Computer Science Department, The University of Auckland (2007)
8. Lowe, D. G.: Object Recognition from Local Scale-Invariant Features, In: Proc. *ICCV* (1999), 1150–1157
9. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision, In: Proc. *IJCAI* (1981) 674–679
10. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions, In: Proc. *British Machine Vision Conference*, volume 1 (2002) 384–393
11. Mikolajczyk, K.. *Detection of local features invariant to affine transformations*, PhD hesis, Institut National Polytechnique de Grenoble, France (2002)
12. Shi, J. and Tomasi, C.: Good Features to Track, In: Proc. *IEEE Conf. Computer Vision Pattern Recognition* (1994) 674–679
13. Tomasi, C. and Kanade, T.: Shape and Motion from Image Streams under Orthography: a Factorization Method, In: *International Journal of Computer Vision*, volume 9 (1992) 137–154
14. van Vliet, L. J., Young, I. and Verbeek, P.: Recursive Gaussian derivative filters In: Proc. *Int. Conf. on Pattern Recognition* (1998), 509–514
15. Vedula, S., Baker, S., Rander, P., Collins, R., and Kanade, T.: Three-Dimensional Scene Flow, In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2005) 475–480
16. Wedel, A., Rabe, C., Vaudrey, T., Brox, T., Franke, U., Cremers, R.: Efficient Dense Scene Flow from Sparse or Dense Stereo Data, Technical report, Computer Science Department, The University of Auckland (2008)