

Learning Motion from Temporal Coincidences

Christian Conrad¹(✉) and Rudolf Mester^{1,2}

¹ Visual Sensorics and Information Processing Lab (VSI),
Computer Science Department, Goethe University, Frankfurt, Germany
`conrad@vsi.cs.uni-frankfurt.de`

² Computer Vision Laboratory, Electrical Engineering Department (ISY),
Linköping University, Linköping, Sweden
`mester@isy.liu.se`

Abstract. In this work we study unsupervised learning of correspondence relations over extended image sequences. We are specifically interested in learning the correspondence relations ‘from scratch’ and only consider the temporal signal of single pixels. We build on the Temporal Coincidence Analysis (TCA) approach which we apply to motion estimation. Experimental results showcase the approach for learning average motion maps and for the estimation of yaw rates in a visual odometry setting. Our approach is not meant as a direct competitor to state of the art dense motion algorithms but rather shows that valuable information for various vision tasks can be learnt by a simple statistical analysis on the pixel level. Primarily, the approach unveils principles on which biological or ‘deep’ learning techniques may build architectures for motion perception; so TCA formulates a hypothesis for a fundamental perception mechanism. Motion or correspondence distributions as they are determined here may associate conventional methods with a confidence measure, which allows to detect implausible, and thus probably incorrect correspondences. The approach does not need any kind of ground truth information, but rather learns over long image sequences and may thus be seen as a continuous learning method. The method is not restricted to a specific camera model and works even with strong geometric distortions. Results are presented for standard as well as fisheye cameras.

1 Introduction and Related Work

Estimating the relation between images, e.g., from multiple cameras or between subsequent frames in a video sequence is one of the most fundamental tasks in visual pattern recognition. Taking a biology-oriented view on vision, we may ask how processing architectures that successfully unveil the inherent structure of a video signal can evolve almost automatically. Conventional approaches to correspondence estimation between images are almost exclusively based on detecting, matching, or tracking *spatial features*. In this paper, we take a complementary

C. Conrad and R. Mester—This work was in parts supported by the ELLIIT programme funded by the Swedish Government.

position and analyse the relation between images by looking at the *temporal* course of single pixel signals. It has already been shown previously [3] that by employing this approach (Temporal Coincidences Analysis, TCA), it is possible to reliably estimate the geometric and photometric [4] relations between images taken from a set of arbitrarily oriented cameras looking at the scene from very different view points [5], as well as for a moving stereo rig [3]. It is important to note that our approach is not applicable to single image pairs but requires a long stream of images. Therefore, our method targets vision systems in robotics, driver assistance and other multi-camera systems where long image streams are naturally available from which the regarded system can/should autonomously learn. This learning process can be scaled down almost arbitrarily in time (while increasing learning time) and thus can act as a lightweight subordinated task on a vision system.

In the present paper, we address motion *within* a monocular stream of images taken from a (not necessarily) moving camera. There is an abundance of algorithms to estimate optical flow, ranging from local to global methods such as the classical work by Lucas and Kanade [13] and Horn and Schunck [9]. Since then, the basic idea of brightness constancy and/or locally constant motion have been used and extended many times to design highly optimised algorithms (see [1, 16] for an overview). However, these methods do not explain how motion perception may evolve and may be learnt over time. We emphasise strongly that this paper is not about computing individual motion vector fields connecting two particular images of a sequence, but about estimating the *statistics* of such motion vector fields without explicitly computing motion vectors. Furthermore, we address the problem of extracting meaningful *hidden variables* which have a large importance in describing the overall structure of a complex spatio-temporal pattern. In the present paper, we specifically look at variables which are closely coupled to the perception of lateral ego-motion. Learning of motion patterns (persistent motion) has found attention in the computer vision community, especially for video surveillance. Here, the goal is to model and extract persistent or normal motion patterns, which may be used to detect abnormal behaviour or may be used as a prior in a tracking application. Motion patterns are extracted based on the 3D structure tensor [18], from object tracks [7], or based on clustered sparse or dense optical flow fields [10]. Compared to these approaches, we never build object tracks and we never explicitly compute the optical flow. Instead, we apply TCA and solely rely on the temporal difference of single pixels and aggregate motion/correspondence candidates over time.

In the vision community, learning optical flow is mainly addressed by estimating the parameters of a specific model from ground truth data. The focus clearly lies on designing methods that advance the state-of-the-art by means of accuracy and/or running time on, e.g., the Middlebury benchmark [1]. Sun et al. [17] learn a statistical model of the spatial properties of optical flow by learning the parameters of the model based on ground truth data. They show that the model captures the statistics of optical flow and outperforms several standard methods. In contrast to this, we are interested in exploring how the statistics of

optical flow may be learnt from scratch without any supervision or ground truth. Roberts et al. [15] learn a subspace of dense optical flow and show how the learnt subspace may be used to infer a dense motion map from sparse measurements and to estimate the ego-motion of a moving platform. Based on [15], Hardtweck and Curio [8] estimate the platform heading from monocular visual cues. They infer the *Focus of Expansion* (FoE), and require sparse flow and incremental platform motion for training. The FoE is computed based on the divergence of the optical flow field. In contrast to this, we present experimental results for inferring the hidden variables proportional to the yaw rate of a moving camera without the need for platform motion data.

To summarise, we will present the applicability of the Temporal Coincidence Analysis approach for learning statistics of image motion. The method can adapt to the computational power available while being easily parallelised if desired.

2 Temporal Coincidence Analysis of Motion

The fundamental idea of the Temporal Coincidence Analysis approach for correspondence estimation builds upon the observation that the short-time temporal signature of *corresponding* pixels in different images should be similar. Signatures need to be significantly above usual noise fluctuations in order to be associable across images. In [3] Conrad et al. show how pixel correspondences may be learned in a binocular camera setup by a *Temporal Coincidence Analysis* (TCA), i.e., by the detection and matching of signal changes (temporal coincidences) from long image sequences. As in a spatial approach, matching is done based on features with high information content. In TCA, rare signal changes (events) are detected at a regarded pixel location and subsequently are matched with similar signal changes occurring at a different pixel location. The detection and matching of events is performed over many frames and accumulated; this way the empirical distribution of the correspondence relation is obtained. This correspondence distribution, in turn, can be characterised by a set of attributes, such as its first and second order statistics and thus allow to assess the learning progress in a statistically principled way.

Temporal coincidence analysis is not restricted to the multi-camera or stereo case, but is also applicable when the correspondence relation between pixels in the *same* video stream is explained by optical flow. For reasons that will be explained later, in general only the *average* correspondence relation, or the *distribution* of that correspondence may be learned by TCA. However, experimental results will show that the principle of TCA may also be used to estimate differential motion parameter.

2.1 Approach

We regard monocular image sequences $I_t \in \mathbb{R}^{m \times n}$, $t = 0, \dots, T$, where corresponding pixels between two time steps are encoded by means of the optical flow. For a fixed 2D pixel location \mathbf{s} (*seed* pixel) and time t , an ‘*event*’ is fired if the grey value difference between time $t - 1$ and t is above some event threshold T_e :

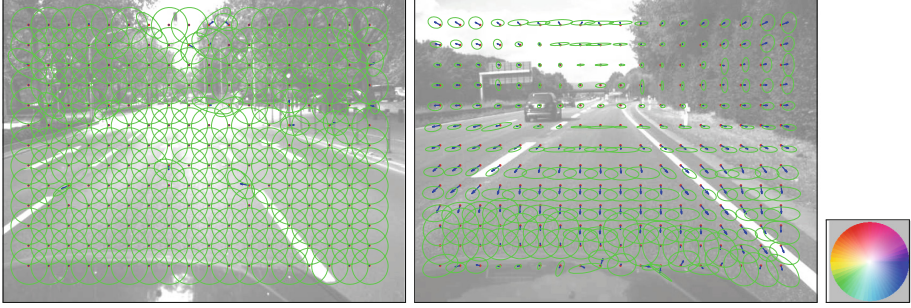


Fig. 1. Learnt flow vectors with associated confidence and color wheel: Sparse flow (blue arrows) and corresponding confidence encoded by means of a (green) covariance error ellipse, at (left) initialisation with maximum uncertainty and (middle) after having processed 24000 frames for sequence **GUCar**. (right) Color wheel used to visualise direction and magnitude of 2D vectors (cf. [1]). Best viewed in color. (Color figure online)

$$|I_t(\mathbf{s}) - I_{t-1}(\mathbf{s})| \geq T_e. \quad (1)$$

Next, we identify those pixel locations which show a similar grey value change within time t and $t + 1$. Typically there will be a set of pixels showing a similar event, and without taking further information into account it is in general not possible to identify the single true correspondence between two time steps. Instead we have a set of possibly corresponding pixels which showed a similar event:

$$\begin{aligned} \Omega_{s_t} = \{ \mathbf{y} \in I_{t+1} : & |I_t(\mathbf{s}) - I_{t-1}(\mathbf{s})| \geq T_e \\ \wedge \quad & |(I_{t-1}(\mathbf{s}) - I_t(\mathbf{y}))| < \epsilon \quad \wedge \quad |(I_t(\mathbf{s}) - I_{t+1}(\mathbf{y}))| < \epsilon \}, \end{aligned} \quad (2)$$

where $\epsilon = 2\sigma$ accounts for the noise standard deviation. The set of possible correspondences per time step are add up in one accumulator array $\mathbf{A}_{\mathbf{s}}$ per seed pixel \mathbf{s} . If nothing is known about the scene layout, the true correspondence may be located at any valid pixel location. Thus, in general the accumulator would need to be of the same size as the image itself. However, the range of typical motion is usually known and it is almost always much smaller than the overall image size. The search area, i.e., the area in which events are matched may therefore be restricted to a window of size $w_x \times w_y$ centred on the regarded seed pixel with $\mathbf{A}_{\mathbf{s}} \in \mathbb{R}^{w_x \times w_y}$. As we accumulate correspondence candidates over time, it becomes clear that in general we may only learn the average correspondence relation and thus the average optical flow at pixel location \mathbf{s} . Imagine a stereo setup where the scene depth does *not* vary. Then the correspondence relation is explained by a disparity map which is constant over time. In this case, the average correspondence coincides with the true disparity. However, given that the scene depth varies, the correspondence relation is given by a time-varying disparity map where correspondences are restricted to lie on the epipolar ray. Given that the scene depth varies, TCA will in general learn parts of the

epipolar rays. For similar reasons, the optical flow observed at a specific scene point typically varies such that only the average flow can be learned. TCA is not meant as a competitor to classic stereo or flow algorithms tailored to return instantaneous (per frame) correspondences but may be seen as a prior generator. TCA is an approach to continuously learn and update the distribution of a correspondence relation which allows to guide a higher level process, restrict search areas, or generate confidence information.

2.2 Accumulator Analysis

After a sufficient number of frames have been processed, the accumulator will encode the average correspondence and thus the average optical flow observed within the sequence at location \mathbf{s} . We then extract the most likely correspondence \mathbf{m} as the location of the accumulator mean and compute a confidence measure as the accumulator covariance matrix \mathbf{C} :

$$\mathbf{m} = \begin{bmatrix} m_x \\ m_y \end{bmatrix} = \begin{bmatrix} \sum_x x_i \sum_y \mathbf{A}_{\mathbf{s}}(x_i, y) \\ \sum_y y_i \sum_x \mathbf{A}_{\mathbf{s}}(x, y_i) \end{bmatrix}, \quad (3)$$

$$\mathbf{C} = \begin{bmatrix} \sum_x \hat{x}_i^2 \sum_y \mathbf{A}_{\mathbf{s}}(x_i, y) & \sum_{x,y} \hat{x}_i \hat{y}_i \mathbf{A}_{\mathbf{s}}(x_i, y_i) \\ \sum_{x,y} \hat{x}_i \hat{y}_i \mathbf{A}_{\mathbf{s}}(x_i, y_i) & \sum_y \hat{y}_i^2 \sum_x \mathbf{A}_{\mathbf{s}}(x, y_i) \end{bmatrix}, \quad (4)$$

where the accumulator is assumed to be normalised to sum to 1 with $\hat{x}_i = x_i - m_x$ and $\hat{y}_i = y_i - m_y$. From an eigenvalue analysis we obtain eigenvectors $\mathbf{v}_1, \mathbf{v}_2$ and eigenvalues λ_1, λ_2 with

$$\mathbf{C}\mathbf{v}_1 = \lambda_1\mathbf{v}_1, \quad \mathbf{C}\mathbf{v}_2 = \lambda_2\mathbf{v}_2, \quad (5)$$

where $\lambda_1 \leq \lambda_2$. The eigenvectors encode the direction of highest and lowest confidence scaled by their associated eigenvalue. The eigenvectors define the principal axes of the ‘error ellipse’ which will be used in the following to visualise the confidence about a learnt flow vector. In general, the structure of the accumulator content may be (a) scattered, (b) line like or (c) point like which results in (a) a large error ellipse and high uncertainty in all directions, (b) an oriented error ellipse with high confidence along the direction of eigenvector with smaller eigenvalue and (c) a small error ellipse encoding high certainty in all directions.

In order to visualise the confidence for a *dense* flow map, we adopt a color encoding, similar to the color coding of flow vectors (cf. Sect. 3 for details). Regarding the computational complexity, our method has $\mathcal{O}(n)$ complexity in the number of seed pixels and $\mathcal{O}(n^2)$ complexity in the size of the accumulator where the same holds for the memory consumption.

2.3 Instantaneous Image Motion

Besides learning *average* optical flow, we may also estimate *parameters of instantaneous image motion* by applying TCA in a specific manner. Specifically we show, how the yaw rate (rotation about the downward facing Y axis of the

camera coordinate system, Z axis towards the image plane) may be inferred by analysing the correspondence samples, i.e., the sets Ω_{s_t} available within two time steps for a specific set of seed pixels.

It is well known that the instantaneous image motion vector $(\mathbf{U}(x, y), \mathbf{V}(x, y))^T$ of a general rigid 3D scene can be modeled as [11, 12]:

$$\begin{pmatrix} \mathbf{U}(x, y) \\ \mathbf{V}(x, y) \end{pmatrix} = \begin{bmatrix} \frac{\tau_Z x - \tau_X}{Z} - \omega_Y + \omega_Z y - \omega_Y x^2 + \omega_X xy \\ \frac{\tau_Y y - \tau_Y}{Z} + \omega_X - \omega_Z x - \omega_Y xy + \omega_X y^2 \end{bmatrix}, \quad (6)$$

where $(\mathbf{U}(x, y), \mathbf{V}(x, y))^T$ denotes the image motion vector at location (x, y) , $\tau_{\{X, Y, Z\}}$ denote translational velocities in direction X, Y or Z , and $\omega_{\{X, Y, Z\}}$ denote the angular velocities about the three axis respectively, with $x = X/Z$ and $y = Y/Z$. From Eq. (6) we see that the optical flow at any pixel within the image is a superposition of two independent flow fields: a flow field which only depends on the translational velocities and a flow field which only depends on the angular velocities. Consider the case where the camera pitch and roll rates are zero (or sufficiently small), then from Eq. (6) we obtain:

$$(\mathbf{U}(x, y) \quad \mathbf{V}(x, y))^T = \left[\frac{\tau_Z x - \tau_X}{Z} - \omega_Y - \omega_Y x^2 \quad \frac{\tau_Y y - \tau_Y}{Z} - \omega_Y xy \right]^T. \quad (7)$$

Next, assume that for the translational velocities it holds that $|\tau_{\{X, Y, Z\}}| \ll Z$, then the translational velocities will vanish and we obtain:

$$(\mathbf{U}(x, y) \quad \mathbf{V}(x, y))^T = \begin{bmatrix} -\omega_Y - \omega_Y x^2 & -\omega_Y xy \end{bmatrix}^T \quad (8)$$

$$= \begin{bmatrix} -\omega_Y - \omega_Y (\frac{X}{Z})^2 & -\omega_Y \frac{X}{Z} \frac{Y}{Z} \end{bmatrix}^T. \quad (9)$$

From Eq. (9) we see that (scaled) camera yaw corresponds to the observed image motion of pixels where the pixel's corresponding scene point lies at infinity (or sufficiently far away from the camera):

$$(\mathbf{U}(x, y) \quad \mathbf{V}(x, y))^T = \begin{bmatrix} -\omega_Y & 0 \end{bmatrix}^T. \quad (10)$$

Now we can infer the current yaw rate based on TCA as follows: We detect and match events for those pixels where the corresponding scene point lies sufficiently far away from the camera. From the above derivations we have that the true correspondence for each pixel lying at infinity is given by $-\omega_y$. We then add up all correspondence sample sets Ω_{s_t} and build the *mother accumulator* \mathbf{A}_m for time step $t + 1$. Given that the yaw rate is the dominant image motion, the mother accumulator then shows a distinct peak at the location of the true (but typically scaled) yaw differential. In Sec. 3 we present an approach for selecting pixels at least approximately lying at infinity.

3 Experiments

In this section, we present experimental results of the proposed method and show its applicability for several real world sequences publicly available



Fig. 2. Dense flow and confidence maps: for sequence **GUCar** after 50000 processed frames. (left) Colour coded dense flow map, (middle) eigen vector corresponding to the smaller and (right) larger eigenvalue of the accumulator’s covariance matrix. Colour encoding according to colour wheel (cf. Fig. 1). Best viewed in colour. (Color figure online)

or recorded by us. In all experiments, no ground truth was used; all information has been learnt from scratch. The experiments were performed based on a straight forward `python` implementation. Depending on the computational resources available and the number of seed pixels chosen, the method can operate in real time. In order to recover dense flow fields, the scheme should be applied within a multi scale framework; an initial coarse map is then refined, making use of an optimal accumulator size given by the estimated flow from a previous level. Furthermore the scheme may easily be parallelised. However, for the results presented we make use of a baseline implementation without coarse to fine and/or parallel computations. In the first experiment, we learn dense average flow maps for sequences **GUCar**, **KITTI-Odo**, **GUOmn**i and **Virat**. All sequences are converted to grayscale if needed. Sequence **GUCar** recorded by our group consists of roughly 50.000 frames with a spatial resolution of 640×480 pixels and a temporal resolution of 30 fps. The event threshold was set to $T_e = 20$ with $\sigma = 3$ and the accumulator size was set to $w_x = w_y = 40$ pixels. Figure 1 shows a subset of the learnt flow vectors at initialisation and after having processed 24000 frames, respectively. The average flow for each pixel is extracted as the coordinates of the mean of its associated accumulator. The location of the mean then encodes the actual flow vector learnt and is visualised as a blue arrow in Fig. 1. With each learnt flow vector, a measure of confidence is provided by the accumulator covariance matrix (cf. Sect. 2.2). The confidence is then visualised as a green error ellipse, defined by the eigenvalues and eigenvectors of the accumulator’s covariance matrix. During initialisation, the uncertainty reflected by the ellipse is maximal (Fig. 1 (left)) and becomes smaller as more frames are processed (Fig. 1 (middle)). Figure 2 (left column) shows a dense flow map, learnt after 50000 frames. The direction and magnitude of a flow vector is converted into a color and saturation value respectively, typically given by the ‘Middlebury’ color wheel (shown in Fig. 1 (right)). In the early learning phase (first few hundred frames), the dense flow learnt will usually be noisy and the confidence for large parts of the flow map is low. As the learning continues, the flow map becomes more smooth, the uncertainty decreases and as expected the dense flow

map shown in Fig. 2 converges to the expected average flow field observed by a camera moving through a natural street scene. The orientation of the flow vectors in the left half of the map lie within $[90^\circ, 270^\circ]$ while in the right half the orientation lies in $[90^\circ, -90^\circ]$. Note the area where the flow vectors have a low magnitude. This area corresponds to the average focus of expansion [2], i.e., the location where all points seem to emanate. Figure 2 (middle and right column) visualises the colour coded first and second eigenvector of the accumulator's covariance matrix respectively, scaled by the associated eigenvalue. Throughout all figures the colour coded flow vectors are normalised to the maximal possible flow (as given by the accumulator size). Similarly, the confidence maps are normalised as well. From these confidence maps it can be seen that within the area directly in front of the car, the confidence about the learnt flow vector is rather low in all directions. This is to be expected, as this area corresponds to the safety clearance, where most of the time only the homogeneous road surface is visible and no events are generated. Highest confidences are attained in the upper half of the image, where most of the events are generated. In Fig. 3 we show results for sequence KITTI-0do taken from the KITTI benchmark dataset [6] (specifically sequence 00 from the odometry dataset). The sequences contain 4.500 frames with a resolution of 1246×374 pixels recorded with 30 fps. The parameters of

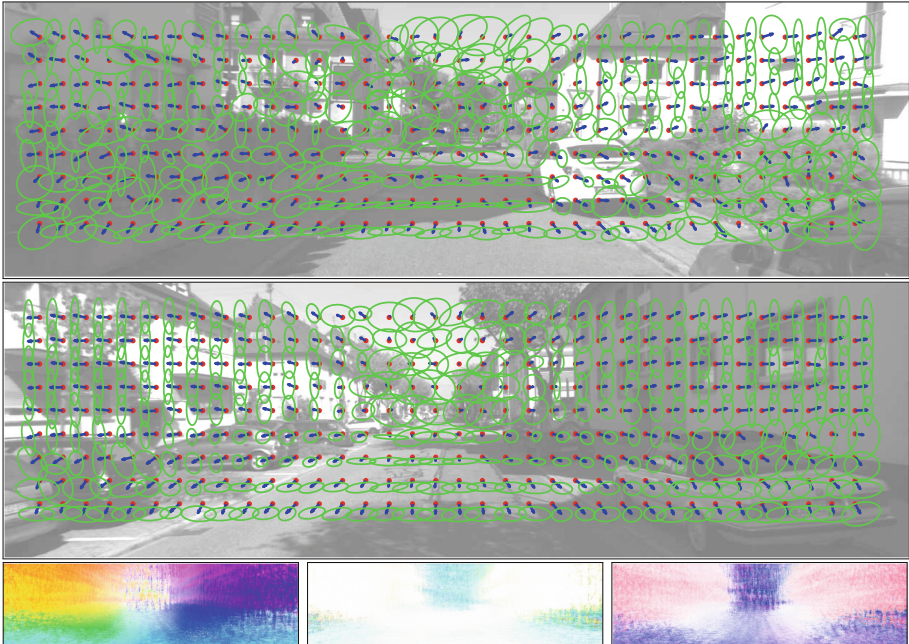


Fig. 3. Learnt flow vectors and associated confidence: Sparse flow and corresponding confidence encoded by means of a covariance error ellipse, learnt after (top) 500 and (middle) 4500 frames for sequence KITTI-00. (bottom) Dense flow map learnt after 4500 frames and associated confidence maps. Best viewed in color and upscaled. (Color figure online)

the method are set as described for sequence **GUCar** before. From the figures, we can basically draw the same conclusions as before. However, the flow maps are considerably more noisy which is due to the low number of frames available.

Figure 4 (top) shows a learnt dense flow map for sequence **GUOmn**i, recorded by our group. The sequence consists of 12.500 frames with a spatial resolution of 640×480 pixels at 30 fps. The camera is equipped with a fish-eye lens and moves through an indoor office environment. It can be seen, that the proposed method may also be used for non standard camera models. The presented approach may of course be applied to static cameras as well. In Fig. 4 (bottom) we show results for sequence **Virat**, taken from the Virat data set [14]. Average flow maps for static cameras are of special interest in surveillance scenarios, where typical and abnormal behaviour is to be detected. To summarise, the first experiment shows that average flow maps can be learnt by just looking at the temporal change of single pixels. These flow maps may subsequently be used as a prior for other flow algorithms. Learning over very long image sequences (not large amounts of ground truth!) clearly is in contrast to the still lasting trend in the vision community to solve a specific problem on short sequences or even still images. The results demonstrate that by only looking at the temporal signal of single pixels useful information may be learnt without any supervision.

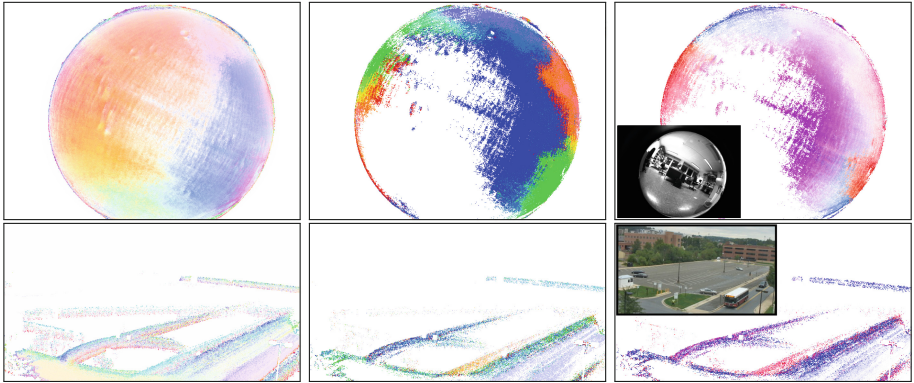


Fig. 4. Dense flow and confidence maps: (top row) (left) Dense flow map learnt after 12500 frames for sequence **GUOmn**i and (middle, right) associated confidence maps (bottom row) Dense flow map learnt after 30000 frames for sequence **Virat** and associated confidence maps. Best viewed in color. (Color figure online)

3.1 Inferring Instantaneous Motion

Next, we show how instantaneous (differential) image motion can be estimated via TCA as well. Recall from Sect. 2.3 that we may infer the yaw rate from a set of pixels lying at infinity. Pixels at infinity are those for which the average flow is small; they can be determined by first learning the average flow map and

the local flow dispersion. However, for the street scenes used in the previous experiment, we can select pixels as follows: We subdivide the image in 4 sectors by two diagonal lines which pass roughly through the focus of expansion. The upper sector can be well used for estimation of the horizontal shift and hence the yaw rate, since the rotational part does not contribute significantly in vertical direction. Figure 5 visualises the selected pixels for some sample images of sequence **GUCar**, where the car turns left, goes straight and turns right. Figure 5 also shows the associated mother accumulators. It can be seen that left and right turns show up as a distinct peak within the accumulator in the opposite direction. To extract the actual yaw rate, we pool the accumulator (=sum up) vertically to obtain a more robust estimate. When vertical motion is not the dominant motion, or if only a few number of selected pixels show an event, the mother accumulator will be scattered. However, this can easily be detected, e.g. by an eigen value analysis as for the normal accumulators. We found that left and right turns can be detected with high confidence (by means of the accumulator statistics as described), given that the analysis is carried out on a sufficiently large number of active pixels. Forward and backward motion can not directly be read off the mother accumulator, as it will in general be scattered for both types of motion. To summarise, this experiment has shown that TCA can also be used to estimate differential motion.



Fig. 5. Inferring current yaw rate: Car turns left (left) goes straight (middle) and (right) turns right. From pixels within the upper triangle the current yaw rate is to be inferred. At each time step only the pixels showing an event are taken into consideration, here marked blue. At each time step a mother accumulator is build by summing up all Ω_{s_t} from which the current yaw rate can be inferred shown in the lower right part of every image. See text for details. (Color figure online)

4 Conclusion

We demonstrated that *temporal coincidence analysis (TCA)* is able to autonomously learn to ‘perceive’ and quantify motion without using any of the elaborated techniques currently used in image analysis. The architecture is able to successfully learn the local mean and dispersion of motion maps and can also be used to perform inference on global ‘hidden variables’ (here: global motion parameters). The method can deal with different camera models in a principled manner. Learning is done completely unsupervised without the need for any ground truth data.

References

1. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. *IJCV* **92**, 1–31 (2011)
2. Ballard, D.H., Brown, C.M.: *Computer Vision*. Prentice Hall, Upper Saddle River (1982)
3. Conrad, C., Guevara, A., Mester, R.: Learning multi-view correspondences from temporal coincidences. In: *CVPR Workshops* (2011)
4. Conrad, C., Mester, R.: Learning relative photometric differences of pairs of cameras. In: *Advanced Video and Signal based Surveillance*. IEEE (2015)
5. Eisenbach, J., Conrad, C., Mester, R.: A temporal scheme for fast learning of image-patch correspondences in realistic multi-camera setups. In: *CVPRW* (2013)
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: *CVPR* (2012)
7. Grimson, E., Stauffer, C., Romano, R., Lee, L.: Using adaptive tracking to classify and monitor activities in a site. In: *CVPR* (1998)
8. Herdtweck, C., Curio, C.: Monocular heading estimation in non-stationary urban environment. In: *MFI* (2012)
9. Horn, B.K., Schunck, B.G.: Determining optical flow. In: *AI* (1981)
10. Hu, M., Ali, S., Shah, M.: Learning motion patterns in crowded scenes using motion flow field. In: *International Conference on Pattern Recognition*, pp. 1–5 (2008)
11. Irani, M., Anandan, P.: Video indexing based on mosaic representations. In: *Proceedings of IEEE* (1998)
12. Longuet-Higgins, H.C., Prazdny, K.: The interpretation of a moving retinal image. *Proc. R. Soc. Lond. B* **208**, 385–397 (1980). <https://doi.org/10.1098/rspb.1980.0057>
13. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. *IJCAI* **81**, 674–679 (1981)
14. Oh, S., Hoogs, A., Perera, A., et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: *CVPR* (2011)
15. Roberts, R., Potthast, C., Dellaert, F.: Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies. In: *CVPR* (2009)
16. Sun, D., Roth, S., Black, M.J.: Secrets of optical flow estimation and their principles. In: *CVPR* (2010)
17. Sun, D., Roth, S., Lewis, J., Black, M.: Learning optical flow. In: *ECCV* (2008)
18. Wright, J., Pless, R.: Analysis of persistent motion patterns using the 3D structure tensor. In: *Workshops on Application of Computer Vision* (2005)

New Trends in Image Analysis and Processing – ICIAP
2017

ICIAP International Workshops, WBICV, SSPandBE, 3AS,
RGBD, NIVAR, IWBAAS, and MADiMa 2017, Catania, Italy,
September 11-15, 2017, Revised Selected Papers
Sebastiano, B.; Farinella, G.M.; Marco, L.; Gallo, G.
(Eds.)

2017, XV, 480 p. 155 illus., Softcover

ISBN: 978-3-319-70741-9