A temporal scheme for fast learning of image-patch correspondences in realistic multi-camera setups

Jens Eisenbach[1], Christian Conrad[1]
[1]Department of Computer Science
Goethe University Frankfurt

Rudolf Mester[1,2]
[2]Department of Electrical Engineering
Linköping University

Abstract

This paper addresses the problem of finding corresponding image patches in multi-camera video streams by means of an unsupervised learning method. We determine patch-to-patch correspondence relations ('correspondence priors') merely using information from a temporal change detection. Correspondence priors are essential for geometric multi-camera calibration, but are useful also for further vision tasks such as object tracking and recognition. Since any change detection method with reasonably performance can be applied, our method can be used as an encapsulated processing module and be integrated into existing systems without major structural changes. The only assumption that is made is that relative orientation of pairs of cameras may be arbitrary, but fixed, and that the observed scene shows visual activity. Experimental results show the applicability of the presented approach in real world scenarios where the camera views show large differences in orientation and position. Furthermore we show that a classic spatial matching pipeline, e.g., based on SIFT will typically fail to determine correspondences in these kinds of scenarios.

1. Introduction

The analysis of multi-camera surveillance data requires techniques very different from both conventional small-baseline stereo as well as from multi-view scene reconstruction. In realistic surveillance scenarios typically no restrictions are imposed on the camera orientation and position, often the viewing directions of the cameras are directly opposite in order to observe objects from different sides.

While the geometric calibration of such a system might be possible, e.g., based on calibration objects or fiducial markers, all manual and supervised approaches to calibration will be exhausting and expensive.

With the ever-growing number and size of camera networks, there is need for automated calibration methods in order to install and run large systems efficiently.

In recent years, methods were developed which determine correspondences not from spatial patterns, but from an analysis of temporal processes in video streams. But also there are two (although different) fundamental problem scenarios that cause a failure in a temporal analysis: a) significant depth differences between corresponding pixels and b) occlusion of objects in one of the views (e.g. with directly opposing viewing directions). The implied assumption is that the cameras are placed very high above ground level, such that the height of moving objects and their associated displacement of corresponding points along the epipolar lines remain relatively low. However, many typical surveillance setups (especially indoor scenes) cannot meet these requirements.

We present a method to determine image-region correspondences for arbitrary multi-camera setups, which we denote as 'correspondence priors'. Those priors encode correspondences on a sub image level by matching a subset of pixels in one view with subsets in other views. The key idea behind our method is to significantly reduce the amount of possible correspondences rather than to achieve (sub-)pixel precision. Then, approaches to match pixel correspondences can exclude an enormous amount of potential corresponding pixels.

We focus on camera setups where a conventional spatial feature matching pipeline will typically fail do determine correspondences, e.g., where the cameras have a large baseline, and significantly differ in scale and orientation. Our approach determines corresponding regions, based on the temporal information of binary change masks. In contrast to other methods using change detection as an essential aspect to determine correspondences [19, 6, 20] we aggregate evidence for correspondence over time which is then statistically analyzed. Most notably, no binary time series are matched like *e.g.* in [6], stabilizing our method w.r.t. falsely labeled pixels (*e.g.* holes in the change mask, camera noise, or jitter).

Our method can be summarized as follows: For (automatically) selected pixels (=seed pixels) in a reference view corresponding cells in a second view are determined based

978-0-7695-4990-3/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPRW.2013.121

on the repeated detection of simultaneous change events among the camera views. With every seed pixel and cell a change probability is associated which is accumulated over time. Correspondences are then extracted from these accumulators based on graph theory. The only assumption that is made is that relative orientation of pairs of cameras may be arbitrary, but fixed, and that the observed scene shows visual activity. While our approach will yield pixel-to-cell correspondence priors, these priors can serve as a precursor to pixel accurate correspondences by massively reducing the search space.

Since any change detection method with reasonably performance can be applied, our method can be used as an encapsulated processing module and be integrated into existing systems without major structural changes.

2. Related Work

The estimation of correspondences among a set of cameras is a central task in multi-camera calibration. However, simplifications of the correspondence problem, e.g., that the orientation, position and physical parameters (focal length, gain and offset) between two views do not differ significantly (*e.g.* in classic stereo matching [16]) is usually not met in surveillance setups. Although correspondences can be extracted manually, such work is time-consuming and expensive. Svoboda *et al.* [18] estimate correspondences semi-manually by tracking a point light source (laser pointer) within the different views. With increasing network size or when pan-tilt-zoom cameras are used, this method becomes unfeasible. Up to now, the typical approach to correspondence estimation is based on the spatial feature matching pipeline where descriptors are generated within each view, and matched across views ([12, 11, 15]). However, matching based on spatial features descriptors fails if the orientations or positions of the cameras are too different (strong deviations from affinity, *etc.*). Therefore, other methods for automatically identifying correspondences are of high interest particularly for the analysis of wide area multi-camera surveillance setups.

Sinha et al. [17] present a method for automated calibration of multiple cameras by means of dynamic silhouettes. Their approach exploits a correspondence constraint between frontier points on the silhouette and epipolar tangents. Lee *et al.* [10] match object trajectories to estimate a global ground plane for the camera network. Wang *et al.* [21] follow a similar approach and analyze activities in uncalibrated camera networks. Makris *et al.* [13] match trajectories in order to connect non-overlapping viewpoints over entry and exit zones. However, matching of trajectories is error prone in cluttered scenes.

In [22] Wexler *et al.* present an appearance based method to estimate the epipolar geometry from multiple image pairs and assume that the scene depth varies smoothly across the images. While being methodologically related to our approach, we do not require the scene depth to vary smoothly and do not rely on appearance which can dramatically change within wide-baseline setups.

There are several approaches to correspondence estimation based on the analysis of change masks. Szlavik *et al.* [19] estimate so called co-motion statistics in overlapping views to determine point correspondences. Their approach can deal with changes in lighting conditions and different camera positions. Van den Hengel *et al.* [20] determine non-overlapping views within large camera networks by identifying camera pairs where the respective change masks are not compatible. Ermis *et al.* [6] build on a binary time series (changed/not changed) per pixel and determine point-correspondences via matching those time series. In [7], these time series analyzed in order to detect abnormal behavior, or rather to match behavior in multi-camera scenes. However, the authors point out the necessity that the cameras have to be placed with sufficient height above the observed scene a requirement typically not met especially in indoor setups. Furthermore the authors post no information concerning the total number of processed frames.

Conrad *et al.* [4] analyze the temporal gray value pattern in selected pixels (seed pixels) to determine pixel correspondences. If there is a significant change of the gray value pattern in a seed pixel, their method looks for similar changes in the other view(s), and it is explicitly only 'rare events' that are considered in this analysis. These measurements can be performed and accumulated for many pairs of frames. For each seed pixel, a spatial distribution of correspondence candidates is determined which reflects the potential variability of correspondences due to depth variations. For this method to function well, the area of interest must exhibit significant object movement, and a substantial number of frames (in the order of 1,000 – 10,000 and more) have to be processed to determine the sought distributions.

The novel method which we present here is inspired by [20, 6, 4, 19, 22] and combines particular aspects thereof. Much like [20], we treat views on a coarse scale (cells); however we do not search cues that *exclude* cell correspondence, but ones which *support* it by coincidence. Binary-valued states of change are determined by exploiting a change detection (similarly to [20, 6]) and coincidences accumulated (as in [4, 19, 22]) for every cell in each time step. The innovation is in the application of sub image scale (image-patches) and an adaptive update and filtering algorithm.

3. Approach

We begin with choosing a set of *seed pixels* $\vec{y}_k, k = 1, .., K$ in the reference view (= view $\mathcal{V}_1$) for which corresponding cells in a second view are to be estimated. Pixels which are located in the vicinity of a seed pixel get

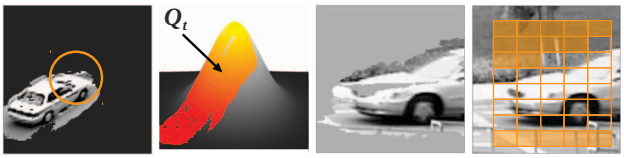

Figure 1. Example of the computation of the change probability and the activation of cells. Left pair: The change mask (propagating car) sets the domain of integration of the Gaussian PDF. Right pair: The number of pixels labeled as changed determine the state of change of each cell. Changed cells are not filled with color.

associated with a weight which corresponds to a normalized Gaussian kernel centered on the seed pixel. The other views $\mathcal{V}_2, \ldots, \mathcal{V}_N$ are divided into equal rectangular cells of dimensions $a \times b$ pixels. The cell size parameters a, b are chosen to be proportional to the expected average size of typical moving objects, say 10-20% of it. The kernel $p(\vec{x}, \vec{y}_k)$ around a seed pixel at position $\vec{y}_k$ is described by

$$p(\vec{x},\vec{y}_k) = \frac{1}{2\pi \cdot |\mathbf{C}|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}(\vec{x}-\vec{y}_k)^T \mathbf{C}^{-1}(\vec{x}-\vec{y}_k)\right), \quad (1)$$

with

$$\mathbf{C} := \begin{pmatrix} \max[a^2, b^2] & 0 \\ 0 & \max[a^2, b^2] \end{pmatrix}. \quad (2)$$

The $\max[\cdot]$ setting ensures circular kernels. For each new time step, a temporal change detection[1] is computed for all views that are analyzed.

Event detection in the reference view $\mathcal{V}_1$: The binary change mask for frame t is multiplied pixel-wise with the normalized Gaussian kernel of each seed pixel $\vec{y}_k$, and the result, being in the interval $[0, 1]$, is denoted as the change probability $\phi_t(\vec{y}_k)$ in the following. If this change probability is larger than an empirical threshold $\gamma_1 = 0.2$, an event has been detected and the seed pixel $\vec{y}_k$ is considered as 'changed'.

Activation of cells In the other views $\mathcal{V}_2 \ldots \mathcal{V}_N$, a cell with total pixel count N_C is considered as 'changed', if more than $N_C \cdot \gamma_2$ pixels ($\gamma_2 = 0.2$) in this cell are changed in the change mask. We denote this process as the 'activation of cells'. This way, a new binary-valued change mask $\mathbf{Z}_t$ which is structured according to the given cell raster is determined. It encodes the changes of cells for two subsequent frames. The computation of the change probability and the activation of cells is illustrated in Fig. 1.

Accumulators for views $\mathcal{V}_2 \ldots \mathcal{V}_N$: If an event is detected in the reference view and if this image region is visible in another view $\mathcal{V}_i$, then at least one cell is expected to change in view $\mathcal{V}_i$ as well. However, in most cases, there will be *multiple* cells in the reference view $\mathcal{V}_1$ which 'fire', and there will also be *multiple* cells in each other view $\mathcal{V}_2 \ldots \mathcal{V}_N$ which fire simultaneously. In the spirit of [4], we denote such co-occurrences of events as 'temporal coincidences'. For each pair of one seed pixel in view $\mathcal{V}_1$ and one other view $\mathcal{V}_i$, a two-dimensional *accumulator array* $\boldsymbol{\mathcal{A}}^i_{k,t}$ is allocated which has as many elements as there are cells in view $\mathcal{V}_i$. While temporal coincidences will also occur between non-corresponding image regions, these false correspondences will not manifest themselves within the accumulator as they do not occur in a systematic way as is the case for the true correspondence.

[1] We employ the statistical model-based method proposed by [2] for reasons of stability. Other state-of-the-art methods can be applied as well.

3.1. Accumulator update rule

The accumulation of temporal coincidences between measured events in the reference view $\mathcal{V}_1$ and changed/activated cells in the other views $\mathcal{V}_i$ is of central importance. For each cell c_{pq}, an accumulator element $(a_{pq})^i_{k,t}$ is provided in which temporal coincidences between events in a seed pixel region (reference view) and changed cells (view 2) are detected and accumulated. This update rule is be based on a) the change probability $\phi_t(\vec{y}_k)$ and b) the context of the accumulator in the previous time step.

For every time step the change probability $\phi_t(\vec{y}_k)$ is computed as described before. To exclude objects too far from the selected seed pixel, a modified change probability is obtained by introducing a lower limit,

$$\tilde{\phi}_t(\vec{y}_i) = \begin{cases} \phi_t(\vec{y}_i), & \text{if } \phi_t(\vec{y}_i) > \gamma_1 \\ 0, & \text{else} \end{cases}. \quad (3)$$

The decision threshold γ_1 is an empirical value and does not influence the functionality of the process. However, the computational complexity can be reduced, as in the case of $\tilde{\phi}_t(\vec{y}_i) = 0$ other views do not have to be checked for which cells have changed. To consider the context of the accumulator, a spatial and time-dependent learning rate $\boldsymbol{\Omega}^i_{k,t} = (\omega_{pq})^i_{k,t}$ is introduced. The learning rate should prefer cells in which or in whose neighborhood many coincidences have been observed. Therefore the accumulator is convolved with a kernel $\mathbf{H}$ according to:

$$\tilde{\boldsymbol{\mathcal{A}}}^i_{k,t} = \boldsymbol{\mathcal{A}}^i_{k,t} * \mathbf{H}, \quad (4)$$

with

$$\mathbf{H} = \begin{pmatrix} 0.05 & 0.15 & 0.05 \\ 0.15 & 0.2 & 0.15 \\ 0.05 & 0.15 & 0.05 \end{pmatrix}. \quad (5)$$

Rescaling the elements of $\tilde{\boldsymbol{\mathcal{A}}}^i_{k,t}$ to the interval $(0, 1]$, the spatial-temporal learning rate $\boldsymbol{\Omega}_{k,t}$ is defined as:

$$\boldsymbol{\Omega}^i_{k,t} = (\omega_{pq})^i_{k,t} := \frac{(\tilde{a}_{pq})^i_{k,t} + 1}{\max[\tilde{\boldsymbol{\mathcal{A}}}^i_{k,t}] + 1}, \quad (6)$$

where $\max[\tilde{\mathcal{A}}^i_{k,t}]$ denotes the maximal value of the accumulator for the time step t. To ensure that no cell has the learning rate value of zero, every accumulator element is increased by one. With the resulting definitions, the update rule of the accumulator is given by

$$\mathcal{A}^i_{k,t} = \mathcal{A}^i_{k,t-1} + \tilde{\phi}_t(\vec{y}_k) \cdot \mathbf{Z}_t \circ \mathbf{\Omega}^i_{k,t-1}. \quad (7)$$

The $\circ$-operator is the Hadamard product (element-wise multiplication of two matrices). Eq. 7 describes the update rule for *one* seed pixel. Thus, for every correspondence a unique accumulator is updated. Only the change mask $\mathbf{Z}_t$ is constant w.r.t. the chosen seed pixel. It only has to be computed once per frame and can be used for all accumulator updates.

Computational complexity Concerning the memory requirements of the method we need to allocate $N \times K \times (a \cdot b)$ integers. Depending on the expected size of a change blob, a and b vary within dozens of pixels. Within a typical multi-camera setup, storage demands are therefore within the megabyte regime.

3.2. Accumulator filtering using a maximum density subgraph

We showed in the previous section how accumulators are updated over time. Next, we describe how to extract those cells from each accumulator which encode the true correspondence with high probability.

In the following we assume that enough temporal coincidences were collected in an accumulator in order to determine a correspondence with a sufficient low uncertainty. Due to noise, erroneous change masks, and discretization in cells etc. many accumulator elements will have a large value, despite they do not encode the true correspondence. Therefore a filter operation is required prior to the extraction of the true corresponding cells. A simple thresholding of each accumulator, e.g., based on the current maximal accumulator value is not feasible because a) in scenes with large depth variation corresponding cells (along the epipolar ray) may be eliminated and b) in scenes with coeval moving objects (*e.g.* traffic scenes) non-corresponding cells will survive. Instead we determine a localized cluster of cells within the accumulator which encode the true correspondence with high probability.

Let K be a subset of accumulator elements a_i , and let $|\vec{r}_{ij}|$ denote the geometric distance (in accumulator units[2]) between two elements of K. Next, we define the non-linear potential function

$$V(K) = \frac{1}{2|K|} \sum_{\substack{i,j \in K \\ i \neq j}} \frac{\sqrt{a_i a_j}}{|\vec{r}_{ij}|}. \quad (8)$$

The purpose of this potential function is to ensure that accumulator elements with a high coincidence matching count contribute strongly to the total potential V. In order to extract the cluster of cells which encode the true correspondence with high probability we determine the set K^* of accumulator elements that maximize Eq. 8. Here the intuition is, that those cells within the vicinity of the true correspondence will have a higher matching count than those far away and will thus form a cluster in accumulator space. K^* is given as:

$$K^* = \underset{K}{\operatorname{argmax}}\, V(K). \quad (9)$$

We cast the optimization problem in Eq. 9 as finding a densest subgraph as follows. Let $G = (V, E)$ be a fully connected undirected graph, where the set of vertices V is given as all elements of an accumulator and the edge weights for pairs of vertices are given by Eq. 8. The density d of such a weighted graph is defined as the ration of the sum of edge weights to the number of vertices [9]. The densest subgraph of G is now found as the graph $G^* = (K^*, E^*)$ maximizing d. While the Goldberg algorithm [9] yields the optimal solution we use a fast greedy approximation developed by Asahiro et al. [3]. The algorithm iteratively removes the vertex with the smallest degree (sum of all edge weights incident to this vertex) and all the edges the vertex is connected to until the graph with desired size is reached. However, this algorithm expects the size of the subgraph as an input parameter which is unknown in our setting. Therefore, we determine the densest subgraph of size $|K| = 2$ based on Asahiro's method and store within each iteration p the density d_p of the current graph of size $|K| - p$. Then the densest subgraph $G^* = (K^*, E^*)$ is the one maximizing d_p. This process is depicted in Fig. 2. Finally, the filtered

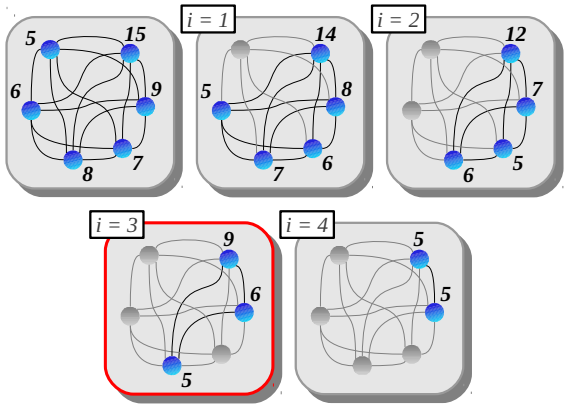

Figure 2. Visualization of the approximation of the densest subgraphs. (top left) The input graph. Numbers besides vertices denote the weighted degree of each vertex, see text for details. Within each iteration i the vertex with the lowest degree is removed and subsequently the density of the remaining graph is determined. Once the densest subgraph of size 2 has been determined, those vertices are chosen, whose induced subgraph has maximum density (red marked frame, $i = 3$). Best viewed in color.

[2] Accumulator units are equal to the corresp. cell width/cell height.

accumulator is given as:

$$\boldsymbol{\mathcal{A}}^*_{k,t} = (a_{pq})^*_{k,t} = \begin{cases} (a_{pq})_{k,t}, & \text{if } (a_{pq})_{k,t} \in K^* \\ 0, & \text{else} \end{cases} . \tag{10}$$

3.3. Back-projection into the image

We interpret the filtered accumulator as an empirical distribution $\alpha(p,q)$ over coordinates p and q and encode the region containing the true correspondence with high probability by means of first and second order moments. Therefore, the filtered accumulator is subject to a normalization as

$$\alpha_{k,t}(p,q) := (a_{pq})^*_{k,t} \; / \; \sum (a_{pq})^*_{k,t} \; , \tag{11}$$

Based on $\alpha_{k,t}(p,q)$ we determine the weighted mean vector $\vec{m}^{\mathcal{A}} = (m^{\mathcal{A}}_p, m^{\mathcal{A}})^T$ and the weighted co-variance matrix $\boldsymbol{C}^{\mathcal{A}}$. These moments are given in accumulator coordinates and are easily converted to image coordinates as

$$\vec{m}^{\mathcal{I}} = (a \cdot m^{\mathcal{A}}_p, b \cdot m^{\mathcal{A}}_q)^T \tag{12}$$

and

$$\boldsymbol{C}^{\mathcal{I}} = \begin{pmatrix} a^2 \cdot \mathrm{Var}[p] & ab \cdot \mathrm{Cov}[pq] \\ ba \cdot \mathrm{Cov}[qp] & b^2 \cdot \mathrm{Var}[q] \end{pmatrix}, \tag{13}$$

where a and b denote the width and height of a cell as already described.

Based on an eigenvalue analysis of the co-variance matrix, we determine so called error ellipses which, depending on the depth variation at the regarded seed pixel, are elongated along the epipolar line w.r.t. to the chosen seed pixel and the principal axis of the objects.

4. Experiments

In the following we present experimental results for various multi-camera benchmark data that show the applicability of the proposed approach in real world setups.

The PETS2006 (Performance Evaluation of Tracking and Surveillance) [1] datasets were of particular interest since they show realistic and challenging surveillance scenarios. Here the scenes show a wide area train station scenario with very different viewpoints and viewing directions which cause significant depth variation between corresponding points.

The Videoweb Activities (VWA) dataset [5] contains several relevant multi-camera sequences; we used an outdoor scene which shows a traffic intersection and an adjacent courtyard. Again, the camera positions vary significantly in orientation and scale of their fields of view. Furthermore, we used several video streams from the LRS work-group at TU Graz [14]; these sequences show a part of the campus with cameras mounted high above ground. Again the views largely differ in orientation and scale.

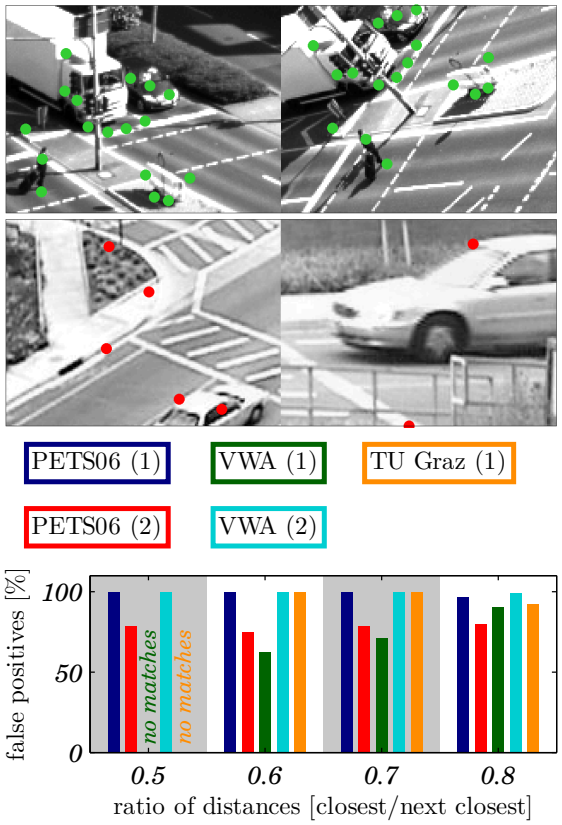

Figure 3. Qualitative and quantitative results of the SIFT framework we used on the sample sequences. *Top left*: Image patches of an urban intersection with relatively similar camera views. SIFT delivers very reasonable results. *Top right*: Intersection sequence from the VWA data set. No points could be matched by a SIFT-based approach in this case. *Below*: False positive rate as a function of the matching metric proposed by [12]. The outlier rate is very high in all sequences, often almost 100%. Matches were hand labeled. Best viewed in color.

We assess the accuracy of the learned pixel-to-patch correspondences in the following way. For each seed pixel $\vec{y}_k$ we extract the first and second order moments from its associated accumulator as described, and determine its co-variance error ellipse. Given that this error ellipse (position given by the mean, size given by the co-variance matrix) contains part of the epipolar ray $\vec{l}_{jk} = \mathbf{F}_j \vec{y}_k$ a correspondence has been learnt. Here $\mathbf{F}_j$ is the fundamental matrix between the reference view and view $\mathcal{V}_j$. We generate ground truth $\mathbf{F}$ matrices for all sequences by hand selected 100 pixel-to-pixel correspondences per camera pair. Subsequently these correspondences are use to robustly estimated the F-matrices via RANSAC [8].

4.1. Results and discussion

To compare the results of our approach with a standard feature matching pipeline, we applied the SIFT framework

to the aforementioned benchmark data. However, for none of the challenging data sets correspondence could be determined in this way. This does not really come at a surprise to us, as matching based on appearance is hard within wide-baseline setups and more importantly where cameras are oriented in opposite directions. Figure 3 shows qualitative and quantitative results based on the SIFT framework.

In contrast to the poor results from a spatial feature matching pipeline, our approach is able to learn correspondences in those challenging surveillance setups. In Fig. 5 for each of the sequences, the learned region correspondences and the associated temporal developments of the accumulators are shown for a subset of 4 chosen seed pixels ($k = 1, ..., 4$). Note that the seed pixels can be selected automatically, e.g., at those pixels where high visual activity is to be expected. The estimated correspondence regions are visualized by means of their respective co-variance error ellipses. For each seed pixel we compute the cumulative change probability as well as the the ratio between the maximum value of the accumulator and the maximum possible accumulator value (= number of events detected at a seed pixel):

$$\Phi_t(\vec{y}_k) = \sum_{\tau=2}^{t} \phi_\tau(\vec{y}_k) \quad \text{and} \quad \max[\tilde{\boldsymbol{\mathcal{A}}}_{k,t}]/\Phi_t(\vec{y}_k). \quad (14)$$

These measures are used to scale the color bars for the visualization of the accumulator at different time steps. Furthermore these numbers are shown right next to each color bar within Fig. 5 and are an indicator for the overall visual activity within the vicinity of the seed pixels.

To better visualize the accuracy of the determined regions, ground truth epipolar lines w.r.t. the chosen seed pixels are shown for every sequence and seed pixel.

In Fig. 5 (rows 1-2), results for sequences from the data set PETS2006 are presented. This scenario features strong variations in both depth and orientation of the cameras. As can be seen from Fig. 5, our method was able to learn the true correspondence regions. Due to the depth variation, these regions align along the epipolar lines w.r.t the chosen seed pixel. At first glance, the blue and yellow regions seem as if they are assigned incorrectly (Fig.5 (row 2)). However, there is no depth information whatsoever of these seed pixels. Accordingly, the process finds *one* possible correspondence along the epipolar line for a seed pixel, which is to be found somewhere between the train and the fence.

Figure 5 (rows 3-4) shows results for two sequences from the VWA data set. The sequences have relatively little activity, which can be seen in the temporal evolution of the accumulators. Only a few cells are activated and the maximum values are smaller. Note that the camera streams are not perfectly in sync and are affected by strong jitter. However, our method was still able to learn the true corresponding regions.

Finally, in Fig. 5 (row 5), results for sequences from the TU Graz data set are shown. The different viewing directions cause depth variations between corresponding pixels, which are not as remarkable as in the PETS2006 data set. All correspondences could be estimated reliable. Only the blue one is shifted along the epipolar line due to depth variations such that the corresponding points on the ground plane are slightly mismatched.

5. Conclusion

In this paper we introduced a novel method that can rapidly and reliably determine correspondence priors for almost arbitrary surveillance setups. The only assumption that is made is that relative orientation of pairs of cameras may be arbitrary, but fixed, and that the observed scene shows visual activity. In extensive experiments, the performance of the method has been evaluated on the basis of publicly available benchmark data-sets. In most cases (86%) the estimated priors were highly precise. Even in the few cases when they were not entirely exact (7%), the results always showed a trend towards the correct result. Sometimes learnt correspondence did not coincide with the true correspondence on the ground plane due to depth variations. The learning approach is unsupervised and the necessary parameters are not critical over a relatively large domain. Furthermore our method can be integrated into existing systems efficiently and may support other computer vision components by information about corresponding regions.

Acknowledgements This work was partially supported by the German Federal Ministry of Education and Research (BMBF) in the project Bernstein Fokus Neurotechnologie – Frankfurt Vision Initiative 01GQ0841, and in parts supported by the ELLIIT, the Lund-Linköping Excellence Initiative of the Swedish Government.

References

[1] www.cvg.rdg.ac.uk/slides/pets.html.

[2] T. Aach, A. Kaup, and R. Mester. Statistical model-based change detection in moving video. *Signal processing*, 1993.

[3] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 2000.

[4] C. Conrad, A. Guevara, and R. Mester. Learning multi-view correspondences from temporal coincidences. In *Proc. CVPR Workshops*, 2011.

[5] G. Denina, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda. Videoweb dataset for multi-camera activities and non-verbal communication. *Distributed Video Sensor Networks*, 2011.

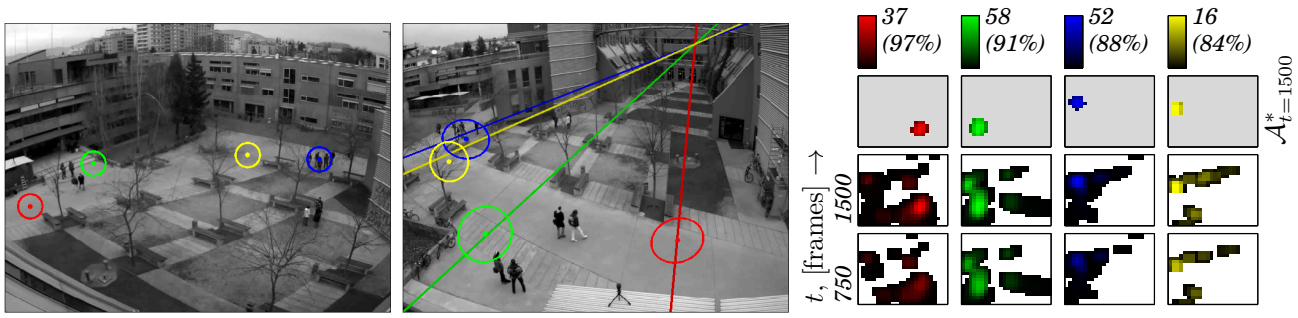

Figure 4. Results for TU Graz(2). Markings as in Fig. 5. Best viewed in color

[6] E. Ermis, P. Clarot, P. Jodoin, and V. Saligrama. Activity based matching in distributed camera networks. *IEEE TIP*, 2010.

[7] E. Ermis, V. Saligrama, P. Jodoin, and J. Konrad. Abnormal behavior detection and behavior matching for networked cameras. In *ICDSC*, 2008.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[9] A. Goldberg. *Finding a maximum density subgraph*. Computer Science Division, University of California, 1984.

[10] L. Lee, R. Romano, and G. Stein. Monitoring activities from multiple video streams: Establishing a common coordinate frame. *IEEE TPAMI*, 2000.

[11] S. Leutenegger, M. Chli, and R. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[13] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *CVPR*, 2004.

[14] H. Possegger, M. Rüther, S. Sternig, T. Mauthner, M. Klopschitz, P. Roth, and H. Bischof. Unsupervised calibration of camera networks and virtual ptz cameras. *CVWW*, 2012.

[15] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *Computer Vision–ECCV 2006*, pages 430–443, 2006.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[17] S. Sinha, M. Pollefeys, and L. McMillan. Camera network calibration from dynamic silhouettes. In *CVPR*, 2004.

[18] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators & Virtual Environments*, 2005.

[19] Z. Szlávik, T. Szirányi, and L. Havasi. Video camera registration using accumulated co-motion maps. *ISPRS*, 2007.

[20] A. van den Hengel, A. Dick, and R. Hill. Activity topology estimation for large networks of cameras. In *IEEE AVSS*, 2006.

[21] X. Wang, K. Tieu, and E. Grimson. Correspondence-free activity analysis and scene modeling in multiple camera views. *IEEE TPAMI*, 2010.

[22] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman. Learning epipolar geometry from image sequences. In *CVPR*, 2003.

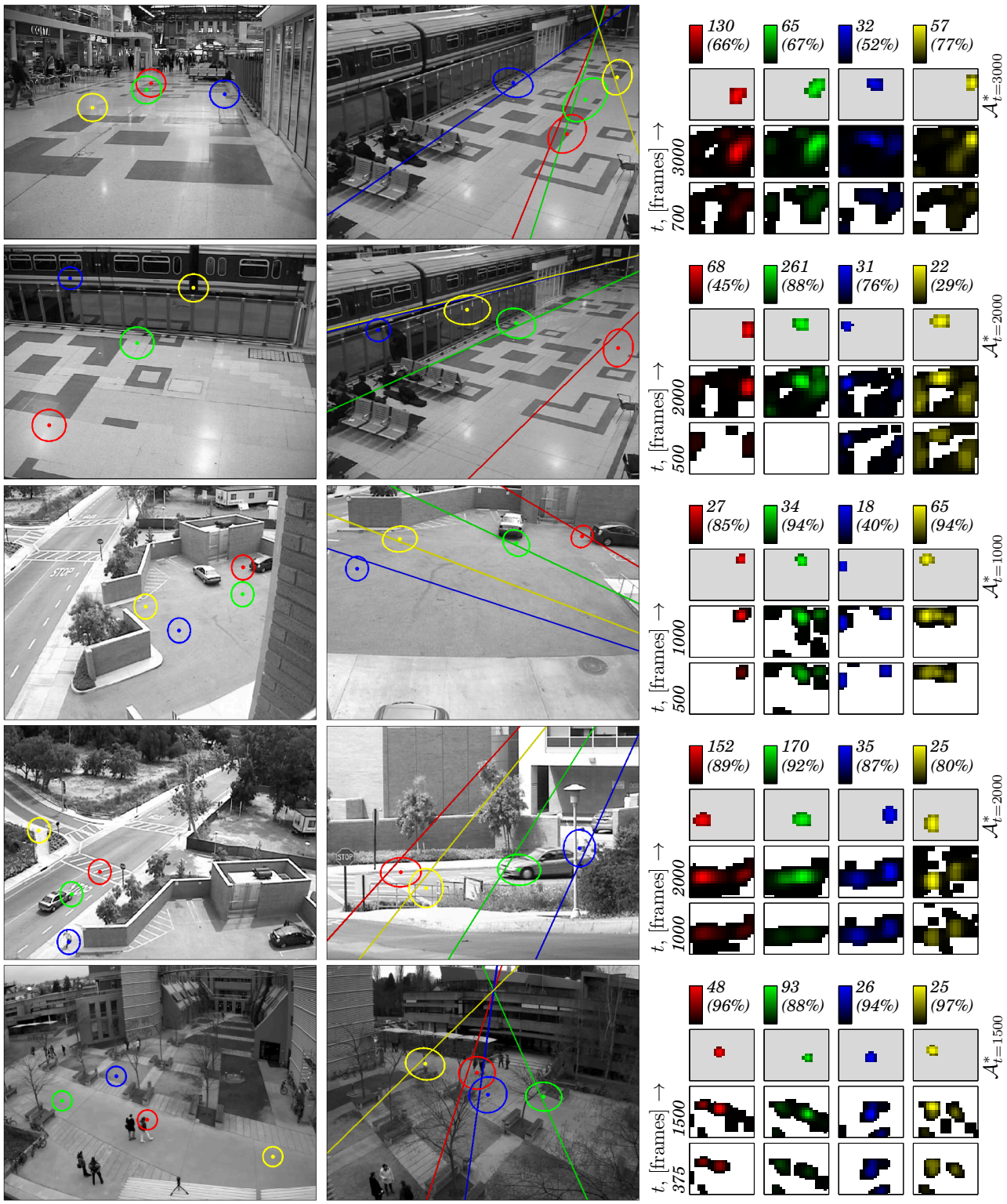

Figure 5. Results obtained by our method for all processed video data. *From left to right column*: Reference view and selected seed pixels, estimated corresponding regions in the second view and temporal developments of the accumulators. Here, the filtered accumulators after the last time step are highlighted in gray. *From top to bottom row*: VWA(1), VWA(2), PETS(1), PETS(2) and TU Graz(1). See subsection 4.1 for details. Best viewed in color.