MULTI-VIEW OBJECT MATCHING AND TRACKING USING CANONICAL CORRELATION ANALYSIS

Marin Ferecatu *Hichem Sahbi*

Institut TELECOM, TELECOM ParisTech
CNRS LTCI, UMR 5141
46, Rue Barrault, 75634 Paris Cedex, France

ABSTRACT

Multi-view tracking of objects in video surveillance consists in segmenting and automatically following them through different camera views. This may be achieved using geometric methods, e.g. by calibrating camera sensors and using their transformation matrices. However, in practice the precision of calibration is a major issue when trying to achieve this task robustly.
In this paper, we present an alternative framework for multi-view object matching and tracking based on canonical correlation analysis. Our method is purely statistical and encodes intrinsic object appearances while being view-point invariant. We will show that our technique is (i) easy-to-set (ii) theoretically well grounded and (iii) provides robust matching and tracking results for traffic surveillance.

Index Terms— Canonical correlation analysis, object matching and tracking, video surveillance.

1. INTRODUCTION

Multi-view object matching and tracking for video-surveillance received a lot of attention in the recent years, motivated by security applications and by the needs of video content providers [1, 2]. This is achieved by extracting objects on individual cameras and matching them across different view-points. The first step is usually achieved using statistical object modelling, motion estimation [3, 4] and spatio-temporal segmentation [5, 6]. Our focus in this work is mainly on the second step, i.e., on multi-view object tracking, so we assume that information about object locations is available for individual cameras.

Without loss of generality, we deal in this work with the case of two *un-calibrated* camera sensors; extension to multiple sensors is straightforward. Our goal is to find, for each time stamp (denoted by t), a set of correspondences among candidate objects and hence trace their trajectories through different views[1] (see Fig. 1). Sensor views are assumed overlapping in order to make the search of candidate matches and hence object tracking possible.

The objects are visually described using a bag of low level SIFT keypoints [7] extracted inside the underlying bounding boxes. Given two overlapping and synchronous video sequences, let n be the number of frames such as the interest object is visible in both cameras and let $\mathcal{L} = \{x_1, ..., x_N\}$ and $\mathcal{R} = \{y_1, ..., y_N\}$ be the set of keypoints extracted from all the bounding boxes in the left and right views; we assume that $\mathcal{L}$, $\mathcal{R}$ are ordered so $x_i \in \mathcal{L}$ will match $y_i \in \mathcal{R}$, i.e., the underlying (2D) SIFT interest points belong to the same physical object. We use canonical correlation analysis (CCA) in order to learn transformations which maximize the expected correlation of pairwise data in $\mathcal{L}$, $\mathcal{R}$ into a common latent space (see §2). We then use these transformations in order to infer (new) object matches on new video sequences of the same scene. Only pairs with the highest correlations are kept as matches. Since the transformation between two camera views might not be linear (due to occlusion, sensor issues, illumination, etc.), the standard linear CCA might not be sufficient. We will show indeed that the extended, non-linear, version of CCA achieves better performances and implicitly handles these transformations between different view-points (see §4).

Instead of CCA, one might consider homographic transformations in order to find matching points using the underlying 2D coordinates (see for example [8]). Regardless of the difficulties linked to camera calibration, this approach fails when camera views contain overlapping bounding boxes because point correspondences based on 2D positions are not discriminative enough. Besides, our CCA-based approach only requires object associations which are easier to setup than the precise point matching required by geometry-based techniques.

The paper is organized as follows: in the next section we review CCA and its kernelized version which allows us to learn object transformations and perform matching through different view-points. In §3 we present our object tracking strategy based on stochastic voting which, given a right choice of parameters, achieves almost perfect tracking results when compared to frame-based tracking. We show the performance of our framework in §4 and we conclude the paper in §5 with a discussion and final remarks.

2. VIEW-POINT TRANSFORMATION LEARNING

Let $\mathcal{X}$ be the input space (for instance the 128 dimensional Euclidean SIFT space) and consider $\mathcal{L}, \mathcal{R} \subseteq \mathcal{X}$, the two sets of training keypoints described in the previous section. The goal of this section is to learn transformation matrices $\mathbf{P}_\ell$, $\mathbf{P}_r$ which make it possible to characterize points in these sets while being view-point invariant.

2.1. Canonical Correlation Analysis

Canonical correlation analysis finds two sets of orthogonal axes in $\mathcal{X}$ such that the projection of data in $\mathcal{L}, \mathcal{R}$ maximizes their correlation and most of their statistical variance. Let $\mathbf{P}_\ell$, $\mathbf{P}_r$ denote the projection matrices of these orthogonal axes which respectively correspond to the left-hand and the right-hand sides camera sensors. CCA finds these matrices by maximizing the following criterion [9, 10] :

$$\begin{array}{rcl} (\mathbf{P}_\ell^*, \mathbf{P}_r^*) & = & \arg\max_{\mathbf{P}_\ell, \mathbf{P}_r} \mathbf{P}_\ell' \, C_{\ell r} \, \mathbf{P}_r \\ \text{s.t.} \quad \mathbf{P}_\ell' \, C_{\ell\ell} \, \mathbf{P}_\ell & = & 1 \\ \mathbf{P}_r' \, C_{rr} \, \mathbf{P}_r & = & 1, \end{array} \quad (1)$$

This work was supported by the French National Project Infomagic.

[1] These views will be referred to as the left and the right side views.

where $C_{\ell r}$ (resp. $C_{\ell\ell}$, C_{rr}) are the interclass (resp. intraclass) covariance matrices of data in $\mathcal{L}$, $\mathcal{R}$. One can show (see for instance [10]) that (1) is equivalent to solving the following eigenproblem

$$\begin{array}{lll} C_{\ell r}C_{rr}^{-1}C_{r\ell}\mathbf{P}_\ell & = & \lambda^2 C_{rr}\mathbf{P}_\ell \\ \mathbf{P}_r & = & \frac{1}{\lambda}\, C_{rr}^{-1}C_{r\ell}\mathbf{P}_\ell \end{array} \quad (2)$$

As already discussed in §1, view-point transformations might not be only geometric as they include other physical aspects including (illumination changes, etc.), so one should consider a non linear version of CCA using kernel mapping (see §2.2 and §4). Prior to describe our matching strategy in §2.3, we will review kernel mapping via kernel principal component analysis (KPCA) in §2.2. The latter makes it possible to control dimensionality of data and helps defining new mapping spaces where CCA transformations become non-linear.

2.2. Kernel Mapping

Let Φ be an implicit mapping (defined via a kernel function $K(x, y) = \Phi(x)'\Phi(y)$) from the input space $\mathcal{X}$ into a high dimensional feature space $\mathcal{H}$. Assume the training set $\mathcal{L}$ is centered in the mapping space $\mathcal{H}$, i.e., $\sum_{i=1}^N \Phi(x_i) = 0$. KPCA finds principal orthogonal projection axes by diagonalizing the covariance matrix $M = (1/N)\sum_{i=1}^N \Phi(x_i)\Phi(x_i)'$. The principal orthogonal axes, denoted $\{V_k, k = 1, ..., N\}$, can be found by solving the eigenproblem $MV_k = \lambda_k V_k$, where V_k, λ_k are, respectively, the k^{th} eigenvector and its underlying eigenvalue. It can be shown (see for instance [11]) that the solution of the above eigenproblem lies in the span of the training data, i.e., $\forall k = 1, ..., N,\ \exists \alpha_{k1}, ..., \alpha_{kN} \in \mathbb{R}$ s.t. $V_k = \sum_{j=1}^N \alpha_{kj}\Phi(x_j)$, where $\alpha_k = (\alpha_{k1}, ..., \alpha_{kN})$ are found by solving the eigenproblem $K\alpha_k = \lambda_k \alpha_k$. Here K is the Gram matrix on the centered data in $\mathcal{L}$ (resp. $\mathcal{R}$) in the feature space. In case the data are not centered, this matrix is defined as

$$K_{ij} = \left\langle \Phi(x_i) - \frac{1}{N}\sum_k \Phi(x_k), \Phi(x_j) - \frac{1}{N}\sum_k \Phi(x_k) \right\rangle,$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product. Each data $x \in \mathcal{L}$ is mapped into $\psi(x) \in \mathbb{R}^h$, where $\psi(x) = \big(\langle x, V_1\rangle, ..., \langle x, V_h\rangle\big)'$ $(h \ll N)$. The same KPCA mapping process is achieved for data $y \in \mathcal{R}$. CCA is now learned on $\psi(\mathcal{L})$, $\psi(\mathcal{R}) \subset \mathbb{R}^h$.

2.3. Synchronous Frame-Based Object Matching

Given two frames belonging to the same time stamp t, and let $\mathcal{O}_\ell \subset \mathcal{L}$ be an object from the left hand side view and $\{\mathcal{O}_r^i\} \subseteq \mathcal{R}$ the underlying candidate matching objects in the right-hand side view. One finds the corresponding object $\mathcal{O}_r^J \in \{\mathcal{O}_r^i\}$, by minimizing the following criterion:

$$J = \arg\min_i \ \frac{1}{|\mathcal{O}_\ell| \times |\mathcal{O}_r^i|} \sum_{\substack{x \in \mathcal{O}_\ell, \\ y \in \mathcal{O}_r^i}} \Big\| \psi(x)'\,\mathbf{P}_\ell \ - \ \psi(y)'\,\mathbf{P}_r \Big\|_2, \quad (3)$$

where $\|x\|_2 = \sum_j x_j^2$ is the L_2 norm and $|\mathcal{O}|$ denotes the cardinality of $\mathcal{O}$. Let p_1 be the probability of success of this procedure. When repeated through n frames, this random process is seen as a binomial random variable (denoted $X_1 \to \mathcal{B}(n; p_1)$) whose parameter p_1 is the probability, given an object in the left-hand side view, that a good match is found. In practice (see experiments), p_1 does not exceed 0.73, so a better stochastic voting procedure is introduced in §3 and proven to be theoretically well grounded .

3. STOCHASTIC VOTING

Let $\mathcal{O}_\ell$ be an object in the left view and $\{\mathcal{O}_r^i\} \subseteq \mathcal{R}$ the set of candidate matches. Instead of doing matching on individual synchronous pairs separately, a better tracking strategy consists in declaring $\mathcal{O}_r^J$ as a good match if and only if the number of times $\mathcal{O}_r^J$ was chosen as a match (through n frame trials) is bigger than the number of times any other object $\mathcal{O}_r^j$ $(j \neq J)$ is chosen. This procedure is robust and we will show that when n (and p_1) are sufficiently large, it converges with high probability to nearly perfect tracking results.

Consider $\mathcal{O}_\ell$ and $\{\mathcal{O}_r^i\}$ ordered such that $\mathcal{O}_r^1$ (i.e. $J = 1$) is the right match to $\mathcal{O}_\ell$ (according to an existing ground truth). Let X_1, X_2,...,X_m be m binomial random variables ($X_i \to B(n; p_i)$ and m is the maximal number of objects in a given frame, in our experiments $m = 5$). Here X_1 stands for the number of times the good match (i.e. $\mathcal{O}_r^1$) is found after n trials while X_j $(j \neq 1)$ stands for the number of times the wrong match (i.e. $\mathcal{O}_r^j$) is found after n trials too.

Following the above stochastic voting strategy, its probability of success is defined as $P(X_1 > X_2 + ... + X_m)$. Here P is the joint probability distribution of $X_1, ..., X_m$. Now, we provide our main result which allows us under some conditions to lower bound the probability of success when using the voting strategy.

Proposition 1 *Consider* X_1,...,X_m *as* m *binomial random variables with parameters* p_1,...,p_m *respectively. If* $p_1 \in [0, 1]$ *is at least* $-\frac{\log(\delta/2)}{2n} + \frac{1}{2}$, *then*

$$P\Big(X_1 > X_2 + ... + X_m\Big) \geq 1 - \delta \quad (4)$$

where $\delta \ll 1$ *is a fixed error rate.*

Proof 1 *The left-hand side of the above inequality is equal to*

$$\begin{array}{ll} & \displaystyle\sum_{\substack{k_1 + ... + k_m = n \\ k_1 > k_2 + ... + k_m}} P\Big(X_1 = k_1, ..., X_m = k_m\Big) \\ = & P\Big(X_1 > \frac{n}{2}, X_2 + ... + X_m < 1 - \frac{n}{2}\Big) \\ = & P\Big(\displaystyle\sum_i^n Z_i \geq \frac{n}{2}\Big), \ \textbf{here } X_1 = \displaystyle\sum_i^n Z_i,\ Z_i \to \mathcal{B}(1, p_1) \\ = & 1 - P\Big(p_1 - \dfrac{1}{n}\displaystyle\sum_i^n Z_i \geq p_1 - \dfrac{1}{2}\Big) \\ \geq & 1 - 2\exp\Big(-2\,n\,(p_1 - \dfrac{1}{2})^2\Big) \quad \Big(\textbf{by Hoeffding's inequality}\Big) \end{array}$$

The sufficient condition is to choose p_1 *such as*

$$2\exp\Big(-2\,n\,(p_1 - \frac{1}{2})^2\Big) \leq \delta \Rightarrow p_1 \geq -\frac{\log(\delta/2)}{2n} + \frac{1}{2}$$

and when $n \to +\infty$ *and if* p_1 *is at least equal to* $\frac{1}{2}$ *then*

$$P\Big(X_1 > X_2 + ... + X_m\Big) \underset{n\to+\infty}{\longrightarrow} 1 \qquad \square$$

4. EXPERIMENTS

We tested our method on a highway traffic data video segment extracted from the Next Generation Simulation (NGSIM) project [12].

The dataset is built using eight slightly overlapping top views along several highways and roads. We used a 120 seconds fragment (1200 video frames) from an approximate 700-meters, two to three lane arterial segment of Peachtree Street, Midtown, Georgia. For matters of space we present here results only for two cameras (however, we obtain similar results for all camera pairs). The memory requirements of the algorithm are reasonable: we only need to store the current frame for each camera view. Our implementation use Python Imaging Library[2], reaching an average speed of 70 frames per second on a 2.7 GHz Pentium-M processor.

Object track data for each camera were kindly provided by Image Solution Lab — EADS Innovation Works (see Fig. 1 for an example). Movement detection and tracking is based on statistical estimation of background pixels [13], tensor voting [5] and spatio-temporal segmentation [6].

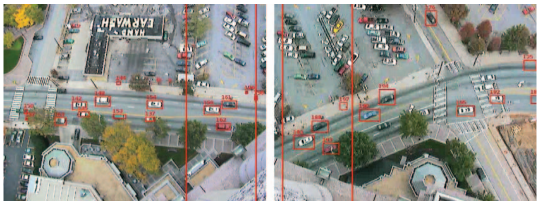

Fig. 1. A two camera view of a highway (traffic video surveillance). Moving vehicles are visible as red bounding boxes; the vertical lines mark the camera overlapping region.

We built a ground truth database of 50 object tracks observed synchronously in both cameras. The overlapping area is manually defined by inspecting the two videos. In our case, since the zone of interest in the image is the highway, two vertical lines are sufficient in order to describe the common region. Each object traverses the overlap area in an average number of 20 frames. We divide the ground truth randomly in two equal parts and use these independently for training and testing. For each frame and each object in the overlap zone we extract the SIFT keypoints using the method described in [7]. For each track in the training set and for each frame we assign pairs of keypoints by visual matching, obtaining the two sets of keypoints ($\mathcal{L}$ and $\mathcal{R}$ in §2) that are used in order to train the CCA transformation.

Given a time stamp t, pairs of "test tracks" are used in order to generate sets of test problems. A test problem consists in fixing an object in one view and finding its corresponding match from the set of objects in the other view. Using this procedure we generated 667 test problems, through different frames, from both directions; left-to-right and right-to-left. For each test problem, the keypoint description of each object is projected (using the matrices $\mathbf{P}_l$ and $\mathbf{P}_r$) into the CCA space. If the input random variables are completely correlated, the two representations should be identical. Since this is not usually the case, the correlation coefficient for each pair of coordinates in the CCA space is slowly decreasing. Since we use the average Euclidian distance to measure the dissimilarity between two objects, we expect that coordinates that correspond to low correlations behave more like noise with respect to the prediction error. We keep thus for further use only the coordinates that correspond to a correlation coefficient larger than a fixed value, denoted by ρ in the following.

First, as a baseline, we test the linear CCA. We use as measure of success the error rate (number of wrong predictions divided by the number of test problems). In Fig. 2 we show the error rate versus the correlation coefficient ρ. As expected, the minimum error ($\epsilon = 0.43$) is obtained for an intermediary value ($\rho = 0.75$). For larger values of ρ too much information is eliminated, thus the error increases while for lower values of ρ, too many noisy (useless) dimensions are decorrelated, also increasing the error rate.

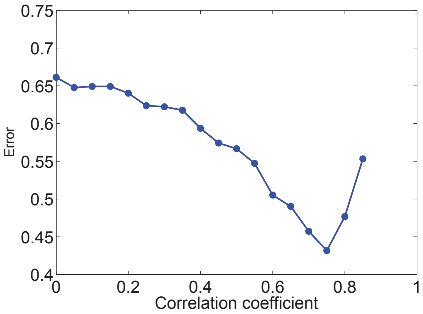

Fig. 2. Prediction error rate versus correlation coefficient ρ for linear CCA. For each value of ρ we keep only the dimensions that correspond to a correlation larger than ρ.

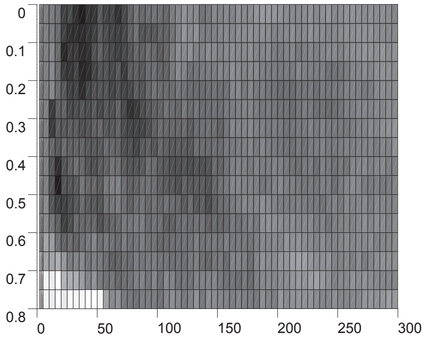

Fig. 3. Prediction error rate for the kernelized version of the CCA algorithm. The vertical axis represents the correlation coefficient ρ and the horizontal axis represents the number of dimensions kept after KPCA. Darker gray values means lower error rates.

Next, we test the kernelized version of the CCA algorithm. We expect to obtain lower error rates; however, there are two parameters to take into account: the correlation coefficient after the CCA step (ρ) and the number of dimensions to keep after the Kernel PCA (denoted by h in §2). We use the Laplacian kernel, $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\, ||\mathbf{x} - \mathbf{y}||)$, with a large value of the scale parameter ($\gamma = 100$). This kernel has been advocated for use in image retrieval [14] and for large values of γ behaves like the triangular kernel, inducing an independence of the results with respect to the scale of data in the description space [15]. In Fig. 3 we present the error rates versus ρ (vertical axis) and h (horizontal axis). As we see, for $h > 150$ the error rates are rather uniform and not particularly small; SIFT features have 128 dimensions: keeping many dimensions does not improve the error rates. We also notice smaller errors (darker grey values) on the main diagonal (top-left to bottom-right). This is explained by the trade-off between the number of dimensions kept after KPCA and the values of ρ; keeping a small number of dimensions

[2] http://www.pythonware.com/products/pil

after KPCA eliminates valuable informations, thus small values of ρ (which does not eliminate further dimensions) provides smaller error rates. Conversely, if a large number of dimensions is kept after KPCA, then many of them will be uncorrelated after CCA and thus a smaller error is expected after eliminating some of them (large ρ). The minimum error rate is achieved for $h = 20$ and $\rho = 0.5$, the underlying sections are shown in Fig. 4 and Fig. 5.

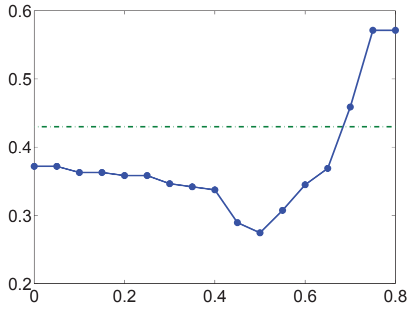

Fig. 4. Prediction error rate versus correlation coefficient ρ for kernelized CCA ($h = 20$). The dotted line represent the best result achieved by the linear CCA (see Fig. 2). The smallest error obtained is 0.27 (compared to 0.43 for the linear version).

Regarding different settings, the best error rate we obtain is 0.27, which implies a probability of successful matches $p_1 = 0.73$. Recall from §3 that the matching procedure is applied on n frames (in our experiments $n = 20$). According to Prop. 1, the event of correct matching (after n frames) occurs with a likelihood larger than $1 - \delta = 0.9998$. The theoretical bound (shown in Prop. 1) is well corroborated as no matching failures are encountered in practice.

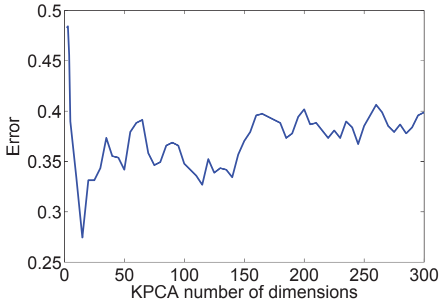

Fig. 5. Prediction error rate versus h for kernelized CCA ($\rho = 0.5$). The best error rate is obtained for $h = 20$. For smaller h too much information is eliminated, while for larger h the added dimensions are not necessarily improving the results.

5. CONCLUSION

We introduced a new object matching and tracking approach for multiple video streams, based on a kernelized version of canonical correlation analysis. Our method uses only the visual descriptions of objects and does not rely on camera calibration or any prior model about scene structures (geometry, etc.) in order to find object correspondences. Experiments conducted on real traffic datasets indicate that the method is highly effective for vehicle matching in multiview sequences. Further work is necessary in order to asses the suitability of the method for cluttered scenes including deformable, moving and varying scale objects; all these rise interesting issues for a future work.

6. REFERENCES

[1] Kideog Jeong and Christopher Jaynes, "Object matching in disjoint cameras using a color transfer approach," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 443–455, 2008.

[2] Shan Y., Sawhney H.S., and Kumar R., "Vehicle identification between non-overlapping cameras without direct feature matching," in *Proc. of the ICCV*, 2005.

[3] Axel Baumann, Marco Boltz, Julia Ebling, Matthias Koenig, Hartmut Loos, Marcel Merkel, Wolfgang Niem, Jan Karl Warzelhan, , and Jie Yu, "A review and comparison of measures for automatic video surveillance systems," *EURASIP Journal on Image and Video Processing*, 2008, doi:10.1155/2008/824726.

[4] R. Venkatesh Babu, Patrick Perez, and Patrick Bouthemy, "Robust tracking with motion estimation and local kernel-based color modeling," *Image and Vision Computing*, vol. 25, no. 8, pp. 1205–1216, 2007.

[5] G. Medioni and P. Kornprobst, "Tracking segmented objects using tensor voting.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[6] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *International Conference on Computer Vision*, 1998.

[7] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] Martin Hofmann Dejan Arsic, Bjorn Schuller, and Gerhard Rigoll, "Multi-camera person tracking and left luggage detection applying homographic transformation," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

[9] Malte Kuss, Malte Kuss, Thore Graepel, and Thore Graepel, "The geometry of kernel canonical correlation analysis," Tech. Rep., Max Planck Institute for Biological Cybernetics, 2003.

[10] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis; an overview with application to learning methods," Tech. Rep., University of London, 2003.

[11] Bernhard Scholkopf, Er Smola, and Klaus robert Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[12] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, "The next generation simulation program," *ITE Journal*, vol. 74, no. 8, pp. 22–26, 2004.

[13] C. Stauffer and W.E.L Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1999.

[14] O. Chapelle, P. Haffner, and V.N. Vapnik, "Support-vector machines for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999.

[15] H. Sahbi and F. Fleuret, "Kernel methods and scale invariance using the triangular kernel," Tech. Rep., INRIA, 2004.