Learning geometrical transforms between multi camera views using Canonical Correlation Analysis

Christian Conrad
conrad@vsi.cs.uni-frankfurt.de

Rudolf Mester
mester@vsi.cs.uni-frankfurt.de

Visual Sensorics and Information Processing Lab, Goethe University
Frankfurt am Main, Germany

Computer Vision Laboratory
Electr. Eng. Dept. (ISY)
Linköping University, Sweden

Abstract

We study unsupervised learning of correspondence relations (point-to-point, or point-to-point-set) in binocular video streams. This is useful for low-level vision tasks in stereo vision or motion estimation as well as in high-level applications like object tracking. In contrast to popular probabilistic methods for unsupervised (feature) learning, often involving rather sophisticated machinery and optimization schemes, we present a sampling-free algorithm based on Canonical Correlation Analysis (*CCA*), and show how 'correspondence priors' can be determined in closed form. Specifically, given video streams of two views of a scene, our algorithm first determines pixel correspondences on a coarse scale. Subsequently it projects those correspondences to the original resolution. For each point in video channel A, regions of high probability containing the corresponding point in video channel B are determined, thus forming correspondence priors. Such correspondence priors may then be plugged into probabilistic and energy based formulations of specific vision applications.

Experimental results show the applicability of the proposed method in very different real world scenarios where the binocular views may be subject to substantial spatial transformations.

1 Introduction

Finding pixel correspondences among a set of views is one of the most important sub problems to be solved in many low-level as well as high-level vision tasks. The basic principle of identifying corresponding pixels or patches appears in many forms: in stereo vision pixel correspondences among a pair of images taken at the same point in time serve to determine a depth map [20, 26]. In motion estimation, pixel correspondences among consecutive images are sought [11, 28]. On a more abstract level, the idea of finding corresponding image patches is also used in object and action recognition (bag-of-X approach [9]).

In the present work, we are particularly interested in identifying pixel correspondences within multi-camera networks. Finding pixel correspondences between pairs of views is here a precursor to the (automatic) determination of camera overlap, in generating a camera graph

http://dx.doi.org/10.5244/C.26.47

(the topographical layout of the camera network), but also within applications like tracking [13]. A different approach to identifying corresponding regions within multiple views is used by van den Hengel et al. in [30] for finding overlapping regions in large camera networks.

In a typical spatial feature approach, one starts with detecting interest points which are subsequently matched based on different metrics to form pairs of potentially corresponding pixels. However, in the prototypical detection and matching framework, correspondences are not determined using the raw pixel representation but based on rich feature descriptors. In the past, feature detectors have often been designed based on statistical (and biological) principles (see [24, 27] for an overview). Today, there is an increased interest in (unsupervised) learning of such features directly from data based on energy-models and probabilistic generative models [7, 17]. Note that unsupervised feature learning is not restricted to model the actual image content but can also be used to learn the relationship or rather: *transformation* between pairs of images [23, 29]. Once the transformation is known, correspondences can be determined in a principled way. While considerable progress has been made in unsupervised learning of features and image transformations, it seems that it is difficult to apply these methods in real world applications. A major drawback is the need for sophisticated optimization schemes, usually involving special sampling schemes as often no closed form solutions exist. Let alone the computational power needed, involving multiple GPUs which makes it hard (or expensive) to use these interesting methods in real world applications.

In this work, we present an unsupervised and sampling-free approach to learn the correspondences relations between pairs of cameras in closed form. The only assumption we make is that the relative orientation between the cameras involved is fixed. Our method is based on a linear model known as Canonical Correlation Analysis (*CCA*). Specifically, given video streams of two views, our algorithm first determines pixel correspondences on a coarse scale via learning the inter-image transformation employing CCA. Subsequently it projects those correspondences to the original resolution. After learning, for each point in video channel A, regions of high probability containing the true correspondence within video channel B are determined, thus forming correspondence priors. Such correspondence priors may then be plugged into probabilistic and energy based formulations of specific vision applications.

1.1 Related Work

Canonical Correlation Analysis (CCA) has been introduced by Hotelling in 1936 [12] as a method of analyzing the relations between two sets of variates. Since then, CCA has been used in quite different disciplines, among them economics, statistical signal processing and climatology [2, 25]. Learning the relationships between paired data is also addressed in the machine learning community, often based on probabilistic generative models [23, 29]. In contrast to such methods, CCA can be applied in closed form, only involving QR decomposition or the SVD, thus making it especially suitable within real world applications. As will be shown later, an important property of CCA is its invariance to affine transformations of the input data. This leads to illumination invariance if the regarded data are patches of image data.

Although there is interesting vision-related work based on CCA, CCA as a tool seems to be familiar to only a part of the (computer) vision community. In [5] Borga and Knutsson combine the phase-based approach to disparity estimation with CCA to estimate depth maps in semi-transparent stereo images. Specifically, CCA is used to generate adaptive linear combinations of quadrature filters in order to be able to estimate multiple disparities at a given image location, which would not be possible when using a standard phase-based approach alone. Furthermore, Borga [4] presented the relationships of CCA and PCA, including other linear models showing that their solutions are given by solving a generalized eigenvalue problem.

Johannson et al. [14] develop a corner orientation detector based on CCA, which is invariant to the actual corner angle and intensity.

Kim et al. apply CCA in high level applications like image and action classification [15, 16]. Specifically, they derive a similarity measure based on CCA within a discriminative learning framework. In [19] Loy et al. develop a framework based on CCA that addresses tasks like activity modeling or finding the spatio-temporal topology within multi-camera networks. While we share a similar application domain, the fundamental difference to our approach is, that they try to identify corresponding regions among non-overlapping cameras that show similar activity. In [8] Donner et al. develop an active appearance model search algorithm based on CCA, where object characteristics are learnt and subsequently used for search.

2 Approach

Our approach to learn correspondence relations and generating correspondence priors between two views is split into two stages. Within the first stage, we learn the inter-image transformation based on CCA. As will become clear shortly, this analysis usually has to be done in a multi-scale framework, depending on the given image resolution, as applying CCA directly to full resolution images may be computationally prohibitive.

In the second stage we employ the learnt transformation which is given only implicitly and show how to predict for a given pixel in a first view its corresponding region within a second view. We denote these regions as correspondence prior.

2.1 Canonical Correlation Analysis

Consider two random vectors $\mathbf{x}$ and $\mathbf{y}$ where $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{y} \in \mathbb{R}^M$. The goal of Canonical Correlation Analysis (CCA) is to find linear combinations (basis vectors) for which the correlation between $\mathbf{x}$ and $\mathbf{y}$ when projected onto the basis vectors are mutually maximized [21]. In the case of a single pair of basis vectors $\mathbf{u}, \mathbf{v}$ the projections are given as $a = \mathbf{u}^T\mathbf{x}$ and $b = \mathbf{v}^T\mathbf{y}$. Using the above definitions and assuming $\mathbb{E}[\mathbf{x}] = \mathbb{E}[\mathbf{y}] = 0$, the (Pearson) correlation (or normalized covariance coefficient) ρ between a and b can be written as:

$$\begin{aligned}
\rho(a,b) &= \frac{\mathbb{E}[ab]}{\sqrt{\mathbb{E}[aa]\mathbb{E}[bb]}} = \frac{\mathbb{E}[(\mathbf{u}^T\mathbf{x})(\mathbf{v}^T\mathbf{y})]}{\sqrt{\mathbb{E}[(\mathbf{u}^T\mathbf{x})(\mathbf{u}^T\mathbf{x})]\mathbb{E}[(\mathbf{v}^T\mathbf{y})(\mathbf{v}^T\mathbf{y})]}} && (1)\\
&= \frac{\mathbf{u}^T\mathbb{E}[\mathbf{x}\mathbf{y}^T]\mathbf{v}}{\sqrt{\mathbf{u}^T\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{u}\mathbf{v}^T\mathbb{E}[\mathbf{y}\mathbf{y}^T]\mathbf{v}}} && (2)\\
&= \frac{\mathbf{u}^T\mathbf{C}_{xy}\,\mathbf{v}}{\sqrt{\mathbf{u}^T\mathbf{C}_{xx}\,\mathbf{u}\mathbf{v}^T\mathbf{C}_{yy}\,\mathbf{v}}}. && (3)
\end{aligned}$$

Note that (3) does not depend on the actual scaling of $\mathbf{u}$ or $\mathbf{v}$, therefore in the case of a single pair of basis vectors CCA can formally be defined as solving the following optimization problem:

$$\underset{\mathbf{u},\mathbf{v}}{max}\ \mathbf{u}^T\mathbf{C}_{xy}\,\mathbf{v} \qquad \text{s.t.} \quad \mathbf{u}^T\mathbf{C}_{xx}\,\mathbf{u} = \mathbf{v}^T\mathbf{C}_{yy}\,\mathbf{v} = 1. \tag{4}$$

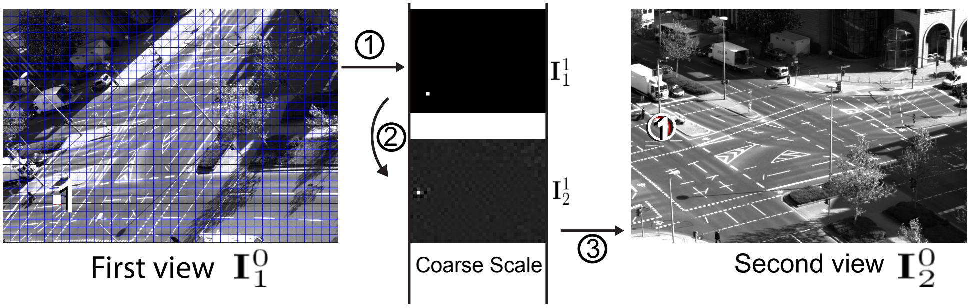

Figure 1: **Generation of correspondence priors:** Visual description of how correspondence priors are generated, once the transformation between two views has been learnt via CCA. See Sec. 2.1.1 for details. Best viewed in color.

Following the derivations of [1] (or similarly [4]) it can be shown that (4) can be cast as a generalized eigenproblem with $K = min(N, M)$ solutions, defining two sets of basis vectors $\{\mathbf{u}_k\}$ and $\{\mathbf{v}_k\}$ with $k = 1, .., K$. Furthermore it can be shown that the projections $a_k = \mathbf{u}_k^T \mathbf{x}$ and $b_k = \mathbf{v}_k^T \mathbf{y}$ are uncorrelated which implies that $\mathbf{u}_i^T \mathbf{C}_{xx} \mathbf{u}_j = 0$ and $\mathbf{v}_i^T \mathbf{C}_{yy} \mathbf{v}_j = 0$ for $i \neq j$. Assuming that $N = M$, and arranging the basis vectors column wise such that $\mathbf{U} = \{\mathbf{u}_k\}$ and $\mathbf{V} = \{\mathbf{v}_k\}$ one can show that $\mathbf{U}$, and $\mathbf{V}$ define a basis for the random vectors $\mathbf{x}$ and $\mathbf{y}$, respectively [1].

In the context of correspondence estimation we form data matrices $\mathbf{X} \in \mathbb{R}^{T \times N}$, and $\mathbf{Y} \in \mathbb{R}^{T \times N}$ where each of the T rows corresponds to a vectorized image $\mathbf{x}^T$, and $\mathbf{y}^T$ of the first and second view, respectively, taken at the same but arbitrary point in time. Without loss of generality we assume that both random vectors have the same dimension ($N = M$).

As will be explained later, it is advantageous to whiten the two data matrices before applying CCA such that $\mathbf{C}_{xx} = \mathbf{C}_{yy} = \mathbb{I}$ holds. Then the two constraints within (4) relax to $\mathbf{u}^T \mathbf{u} = \mathbf{v}^T \mathbf{v} = 1$ and the sets of basis vectors determined by CCA will form orthonormal bases such that $\mathbf{U}^T \mathbf{U} = \mathbf{U}\mathbf{U}^T = \mathbb{I}$ and where the same holds for $\mathbf{V}$. Obviously, to obtain a complete basis the data matrices need to be of full rank (N). Similar as in PCA, in practice CCA can be performed via QR decomposition or SVD of both data matrices without having to estimate covariance matrices from data (see [1, 21] for detailed derivations). However, depending on the application (e.g. when working with smart cameras in a surveillance setup with low storage capacities) and the dimensionality of the input data it may be advantageous to estimate the data covariance matrices instead of having to store large data matrices, as covariance matrices can be estimated online [10].

In summary, it is instructive to think of CCA as a method that tries to register or align pairs of images in such a way that they are maximally correlated. This obviously shares similarities with classical template matching where one identifies pixel correspondences at those locations where the normalized cross correlation is maximum and above some threshold.

2.1.1 Generation of Correspondence Priors

Intuitively, the two basis matrices determined by CCA need to encode the relation or rather transformation between the given views as the correlation between a pair of data will be maximum when they are perfectly registered. However, it may appear more as a theoretical possibility than as an actually feasible approach to learn the transformation between

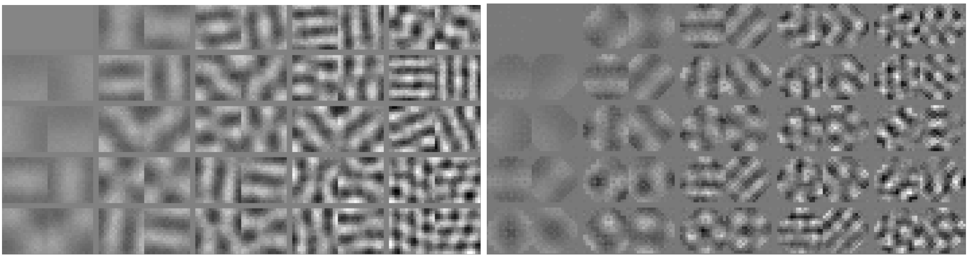

Figure 2: **CCA on natural images:** (left) First 25 pairs of basis vectors obtained when pairs of images are related by a rotation of 90 degrees. (right) First 25 basis vectors obtained when pairs of images are related by a rotation of 45 degrees. See text for details.

paired data via CCA, as explained in the last section. Applying CCA on natural images even for rather low spatial resolutions, say, above 256×256 pixels is computationally prohibitive. Therefore we have to apply CCA within a multi-resolution framework where the transformation between the views $\mathbf{I}_1$and $\mathbf{I}_2$with $\mathbf{I}_1, \mathbf{I}_2 \in \mathbb{R}^{D \times P}$ of a binocular image stream is determined on a coarse scale only. Having learned the transformation on the coarse scale, we can predict for a given pixel on the fine scale in image A which pixel in a low resolution version of image B corresponds to it. Subsequently, the predicted correspondence is backprojected to the original resolution. As correspondence finding on the lower resolution involves a loss of resolution, we cannot find pixel-to-pixel correspondences within the original resolution directly, but rather determine regions which contain the true correspondence with high probability. These regions can then compactly be described with spatial moments (second order statistics) and will be denoted as *correspondence priors*.

Learning the Inter-Image Transformation In the first phase of the scheme, we learn the transformation between the two views as follows. Let T be the number of images in a temporal segment of the binocular image stream. Our algorithm starts with generating the two data matrices $\mathbf{X} \in \mathbb{R}^{T \times N}$, and $\mathbf{Y} \in \mathbb{R}^{T \times N}$ where each row in $\mathbf{X}$ and $\mathbf{Y}$ correspond to a subsampled version of the original image, respectively. During subsampling we keep the aspect ratio of the original resolution such that $D/s \cdot P/s = N$, where s is the subsampling factor.

Recall from Sec. 2.1, that it is advantageous to whiten the two data matrices before applying CCA such that $\mathbf{C}_{xx} = \mathbf{C}_{yy} = \mathbb{I}$ holds. Then the two constraints within (4) relax to $\mathbf{u}^T\mathbf{u} = \mathbf{v}^T\mathbf{v} = 1$ and the sets of basis vectors determined by CCA will form orthonormal bases. As will be explained later, this allows us to perform the prediction in closed form. We whiten the data matrices using PCA while keeping 100% of the data variance (e.g. see [3] pp.568). From this, we obtain matrices $\mathbf{W_x}$, $\mathbf{W_y} \in \mathbb{R}^{N \times N}$ containing the PCA basis vectors for our given data in $\mathbf{X}$, and $\mathbf{Y}$, and matrices $\mathbf{A}_\mathbf{x}^{white}$, $\mathbf{A}_\mathbf{y}^{white} \in \mathbb{R}^{T \times N}$ which hold the projections of the original data onto the respective PCA base. Additionaly each dimension is scaled by the eigenvalues corresponding to the PCA basis vectors. Finally, $\mathbf{A}_\mathbf{x}^{white}$, and $\mathbf{A}_\mathbf{y}^{white}$ contain a decorrelated representation of the original data where each dimension is normalized to have unit variance. Next, we apply CCA on the white data matrices thus obtaining two orthonormal bases $\mathbf{U}_w = [\mathbf{u}_1, .., \mathbf{u}_N]$ and $\mathbf{V}_w = [\mathbf{v}_1, .., \mathbf{v}_N]$, respectively. Obtaining orthonormal bases instead of only orthogonal bases as is the case when applying CCA to non-whitened data (see Sec. 2.1) is particularly useful within the prediction part of our approach as no matrix

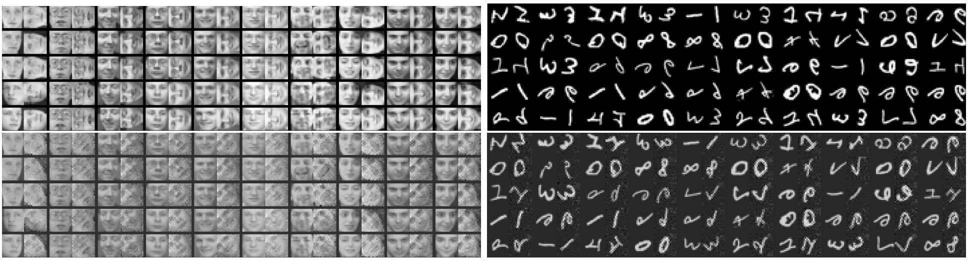

Figure 3: **Applying a learnt transformation to unseen data:** (Top) A learnt transformation of 90 degrees is applied to previously unseen data, here to Olivetti faces and MNIST digits. Each pair of images shows the input image (left) and the result image after applying the learnt transformation (right). (Bottom) As before but for a rotation of 45 degrees. Note how unrelated areas are filled with noise. See Sec. 2.1.1 for details.

inverse needs to be computed.

It is important to note, that the transformation learnt with CCA is given implicitly only. However, in the next section we show how to employ the two basis matrices obtained via CCA in a principled way to predict correspondence regions.

Prediction Given that CCA is able to perfectly learn the transformation then the correlation between a pair of corresponding patches $(\mathbf{x},\mathbf{y})$ will be maximum iff

$$\mathbf{U}_w^T(\mathbf{W}_\mathbf{x}^T\mathbf{x}) = \mathbf{V}_w^T(\mathbf{W}_\mathbf{y}^T\mathbf{y}), \tag{5}$$

which simply means that the representation of $\mathbf{x}$ and $\mathbf{y}$ within their PCA bases $\mathbf{W}_\mathbf{x}$,$\mathbf{W}_\mathbf{y}$ subsequently projected onto the bases $\mathbf{U}_w$, $\mathbf{V}_w$ determined by CCA are identical. This implies that we can predict $\mathbf{y}$ from $\mathbf{x}$ and vice versa which is the same as applying the transformation to $\mathbf{x}$ or $\mathbf{y}$ that relates the patches. Therefore we can predict $\mathbf{x}$ from $\mathbf{y}$ as:

$$\mathbf{U}_w^T(\mathbf{W}_\mathbf{x}^T\mathbf{x}) = \mathbf{V}_w^T(\mathbf{W}_\mathbf{y}^T\mathbf{y}) \tag{6}$$

$$\mathbf{W}_\mathbf{x}\mathbf{U}_w\mathbf{U}_w^T(\mathbf{W}_\mathbf{x}^T\mathbf{x}) = \mathbf{W}_\mathbf{x}\mathbf{U}_w(\mathbf{V}_w^T(\mathbf{W}_\mathbf{y}^T\mathbf{y})) \tag{7}$$

$$\mathbf{x} = \mathbf{W}_\mathbf{x}(\mathbf{U}_w(\mathbf{V}_w^T(\mathbf{W}_\mathbf{y}^T\mathbf{y}))). \tag{8}$$

Within our multi-resolution framework we use the result in (8) to generate correspondence priors as follows. Let (x,y) be the spatial coordinates of a pixel in $\mathbf{I}_1^0$, where the superscript 0 denotes the original resolution. Next, determine the pixels' coordinates within the low resolution as $(u,v) = (x/s,y/s)$, and generate a binary image $I_1^1 \in \{0,1\}^{D/s\times P/s}$ of the same size as the low resolution where the pixel at (u,v) is set to 1 (see step 1 in Fig. 1).

Using (8) we apply the transformation learned by CCA to the binary image and obtain the predicted image $I_2^1 \in \mathbb{R}^{D/s\times P/s}$. Obviously, when there is a one-to-one pixel correspondence and the transformation has perfectly been determined, the predicted image will be binary again where a single pixel is set to 1 marking the corresponding pixel. However, due to noise one typically obtains predictions that encode regions of high probability containing the corresponding pixel (see step 2 in Fig. 1). Interpreting the predicted images as an empirical bivariate correspondence distribution over the spatial image coordinates for a specific pixel we encode the prediction by means of a 2 × 2 covariance matrix $\mathbf{C}_p$. Based on an eigenvalue

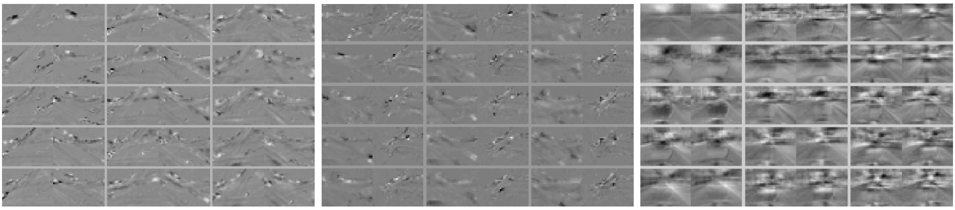

Figure 4: **Learning transformations in real world setups:** First 15 pairs of basis vectors determined by CCA for each of the three sequences (A, B, and C) used within the experiments. See text on page 8 for details.

analysis of $\mathbf{C}_p$, we consider the prediction to be accurate or rather that a correspondence exists if both eigenvalues λ_1, λ_2 of $\mathbf{C}_p$ are small.

Finally, the correspondence prior to the pixel at $(x,y) \in \mathbf{I}_1^0$ in the second view is given as the covariance error ellipse from $\mathbf{C}_p$ projected onto $\mathbf{I}_2^0$ (see step 3 in Fig. 1).

Obviously the prediction only works when the pairs of image patches on which CCA has been performed indeed contains corresponding patches. When applied to unrelated image patches, the prediction will in general not work.

CCA on Natural Images Here, we experimentally verify by means of a simple transfer learning task, that CCA is able to capture the relationship between pairs of images by applying the learned transformation to previously unseen data. We generate two pairs of data matrices by randomly cropping 40 image patches of size 17×17 pixels from each of the 300 images of the Berkeley Segmentation Database [22] which form the first view. Then the second view is generated as (i) the first view rotated by 90 degrees, and (ii) the first view rotated by 45 degrees. Additionally we add i.i.d. zero mean gaussian noise with $\sigma = 3$ separately to each patch in the second view. Next we perform CCA on both pairs of data matrices separately to obtain basis vectors as described before. Figure 2 shows the first 25 pairs of basis vectors obtained by CCA. Note that for a rotation of 90 degrees, there is a one-to-one pixel correspondence between both views, but for a rotation by 45 degrees this is not the case for some pixels. This can also be seen within Fig. 2 (right) where basis vectors develop a circular structure that defines the area visible within both views. Figure 3 shows results when the learned transformation is applied to unseen data, here for faces of the Olivetti database [6] and images of the MNIST dataset [18]. For the case of a 90 degree rotation, there is visually no difference between the original and transformed image. As expected, for the case of a 45 degree rotation regions which are not visible within both views are filled with noise.

In summary, this experiment has shown, that CCA is indeed able to learn basic spatial transformations and that we can apply the learned transformation as described in Sec. 2.1.1, though the transformation is learnt only implicitly.

3 Experiments

Within Sec. 2.1.1 we experimentally verified for pairs of natural images that transformations between paired data can be learned and applied to unseen data based on CCA. Here, we present several experiments that show the applicability of the proposed approach in real world binocular camera setups. Having learnt the transformation between two views, we

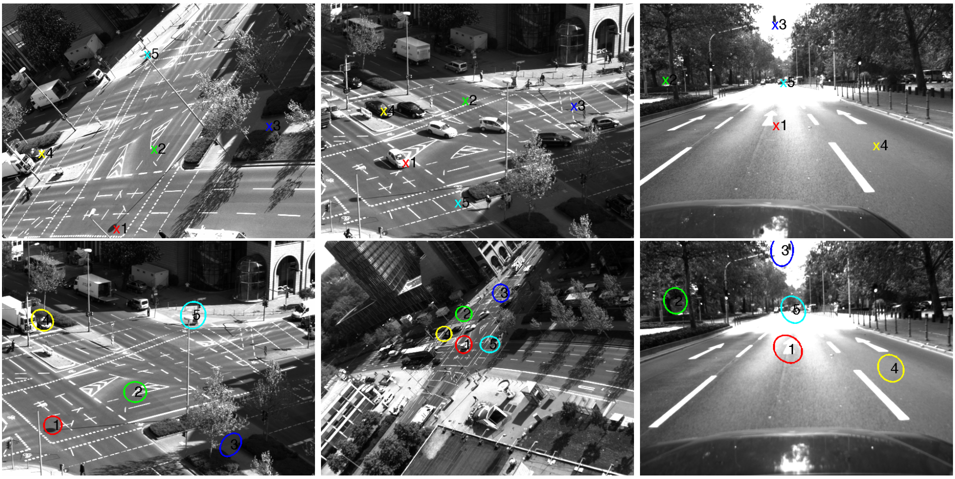

Figure 5: **Correspondence priors for real world setups:** Within each column for sequences A to C: For selected pixels within the first view (top), regions of high probability containing the true corresponding pixel in second view (bottom) are determined based on the proposed approach. See Fig. 1 and text for details. Best viewed in color and upscaled.

show that we can generate correspondence priors or whole correspondence maps that encode visible areas within both views.

Learning Transformations in Real World Setups Using the proposed method we learn transformations between two views for three different settings: (A) static outdoor scene where the cameras are twisted with respect to each other but obey a similar scale (B) static outdoor scene as in (A) where the cameras have different focal lengths thus the views show a large difference in scale and (C) a moving stereo setup where two cameras are mounted on a car that drives through an urban environment. While these datasets are not taken from a standard multi-camera dataset, they represent the setting in which we consider the presented approach to be particularly useful, i.e. in real world multi camera setups with hours or days of video available. While the PETS dataset contains several binocular camera setups, they contain too few data in the range of 2000 to 3000 images only.

For each of the setups both views have a spatial resolution of 480×640 pixels with 10000 consecutive frames per view. We learn the transformations as described in 2.1.1 using a subsampling factor of 16 (with bicubic interpolation), that is, we apply CCA on data matrices $\mathbf{X}, \mathbf{Y}$ of size $10000 \times (30 \cdot 40)$. We perform CCA via SVD on the data matrices which takes roughly 10 seconds on a single 2.6 GHz CPU including the whitening. As already explained, in applications with limited computing or memory resources, CCA can be carried out on estimated covariance matrices. Typically, the covariance matrices have a smaller memory footprint and can be determined incrementally, useful for example in surveillance like applications involving smart cameras.

Figure 4 shows the first 15 pairs of CCA basis vectors (filters) for each of the scenes. It can be seen that for all of the three setups the basis vectors are not random but develop a structure that is characteristic for the setup. Compared to results of unsupervised feature learning approaches that learn image transformations [23, 29], the filters do not develop

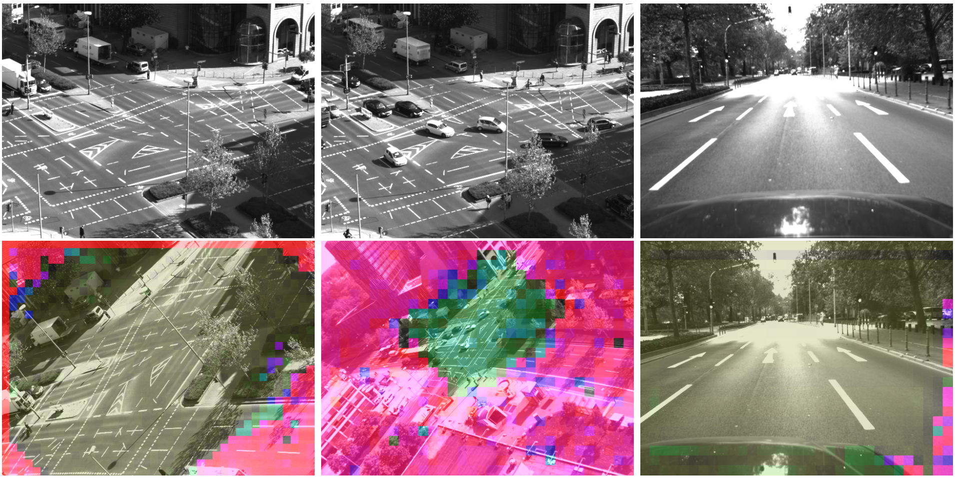

Figure 6: **Correspondence maps for real world setups:** Within each column for sequences A to C: (top) First view of the binocular camera setup. (bottom) Second view overlaid with a confidence map encoding regions of high probability of being visible in both views. Regions colored in shades of purple have a high probability of **not** being visible in the first view. **Results can only be interpreted when viewed in color**.

highly localized Gabor type structures or a Fourier type structure. However, this comes at no surprise, as we do not model a whole class of transformations but rather a single / global transformation that relates the two views.

While the number of image pairs used in general has an influence on the learned transformation, we found the analysis to be rather robust, as long as the data matrices are considerable larger than the theoretical lower bound (see Sec. 2.1). Recall that the absolute number of frames used gives only little insight in how many frames are needed in practice to learn the transformation as the number of *linearly independent* samples is important. Therefore in order to avoid to add many linearly dependent obersavtions to the data matrices, especially in scenes with low activity it is a good idea to not just use N consecutive frames but to exclude subsequent pairs of frames that only differ marginally.

Finding Correspondence Regions Once we have learned the transformation between two views, we generate a correspondence prior in the second view for a given pixel in the first view and vice versa as explained in Sec. 2.1.1. Figure 5 shows examples of such correspondence priors for each of the three different setups (A)-(C). Here it is important to note that correspondence priors can also be generated for pixels which lie within (i) static parts of the scene but more importantly (ii) for pixels which lie within homogeneous parts of the scene. This can be seen in Fig. 5, e.g., selected pixel number 2 in (A) or pixel 4 in (C). Obviously, this is possible since correspondence priors are learned from the spatio-*temporal* coherence or similarity, and not on spatial structure alone.

The correspondence regions shown in Fig. 5 are the covariance error ellipses for a confidence level of 99% containing the true correspondence. If a pixel is not visible within both views, the correspondence prior exhibits high uncertainty indicated by a large error ellipse as both eigenvalues of the priors' covariance matrix would be large.

Finding Overlapping Regions As our approach returns correspondence priors with a confidence measure encoded in a covariance matrix, we can easily find regions visible within both views as follows: For all pixels of one of the two views we generate correspondence priors and store the sum of both eigenvalues of their associated covariance matrix. Observing that the correspondence prior will be the same for all pixels on the original scale which fall within the same pixel on the subsampled view, we do not have to compute $D \cdot P$ correspondence priors but only $D/s \cdot P/s$. This is simply the number of pixels of the subsampled view. Finally, we obtain a confidence map of the size of the original resolution, where low values indicate high confidence in that there is a corresponding pixel in the second view.

Figure 6 shows such confidence maps for all of the three camera setups (A)-(C). This shows that our approach can find overlapping regions within binocular camera setups even when the views are heavily twisted and have a considerable difference in scale.

4 Conclusion

We presented an algorithm to learn the transformation between views of a binocular camera stream in closed form. Furthermore we show how to apply the learned transformation to generate correspondence priors. We consider the approach to be especially useful for real world multi-camera setups with constraints on the computational power and memory footprint. Learning and prediction only involve the QR decomposition (or SVD) and cheap linear transforms. In three different setups we have shown the applicability of the presented method to real world scenarios.

In future work, we plan to use the correspondence priors within a probabilistic framework for motion estimation and distributed object tracking for multi-camera networks.

Acknowledgements This work was 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 programme funded by the Swedish Government.

References

[1] T.W. Anderson. *An introduction to multivariate statistical analysis*. Wiley, 1958.

[2] TP Barnett and R. Preisendorfer. Origins and levels of monthly and seasonal forecast skill for united states surface air temperatures determined by canonical correlation analysis. *Monthly Weather Review*, 115:1825–1850, 1987.

[3] C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.

[4] M. Borga. *Learning multidimensional signal processing*. Linköping University, 1998.

[5] M. Borga and H. Knutsson. Estimating multiple depths in semi-transparent stereo images. In *SCIA*, volume 1, pages 127–134, 1999.

[6] AT & T Laboratories Cambridge. Olivetti face dataset.

[7] A. Coates, H. Lee, and A.Y. Ng. An analysis of single-layer networks in unsupervised feature learning. *AISTATS*, 2011.

[8] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *TPAMI*, 2006.

[9] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, volume 2, pages 524–531. IEEE, 2005.

[10] TF Ghan, GH Golub, and RJ LeVeque. Algorithms for computing the sample variance: Analysis and recommendation. *Amer. Statist*, 37:242–247, 1983.

[11] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 (1-3):185–203, 1981.

[12] H. Hotelling. Relations between two sets of variates. *Biometrika*, 1936.

[13] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank. Principal axis-based correspondence between multiple cameras for people tracking. *PAMI*, 28(4):663–671, 2006.

[14] B. Johansson, M. Borga, H. Knutsson, et al. Learning corner orientation using canonical correlation. In *Proc. SSAB Symposium on Image Analysis*, pages 89–92, 2001.

[15] T.K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *TPAMI*, 29(6):1005–1018, 2007.

[16] T.K. Kim, S.F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR*, 2007.

[17] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368, 2011.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] C.C. Loy, T. Xiang, and S. Gong. Multi-camera activity correlation analysis. In *Proc. CVPR*, pages 1988–1995. IEEE, 2009.

[20] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[21] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate analysis*. Academic Press, 1980.

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, volume 2, pages 416–423. IEEE, 2001.

[23] R. Memisevic. On multi-view feature learning. *ICML*, 2012.

[24] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[25] L.L. Scharf and C.T. Mullis. Canonical coordinates and the geometry of inference, rate, and capacity. *Signal Processing, IEEE Transactions on*, 48(3):824–831, 2000.

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

[27] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151–172, 2000.

[28] C. Stiller and J. Konrad. Estimating motion in image sequences. *Signal Processing Magazine*, 1999.

[29] J. Susskind, G. Hinton, R. Memisevic, and M. Pollefeys. Modeling the joint density of two images under a variety of transformations. In *CVPR*. IEEE, 2011.

[30] A. Van Den Hengel, A. Dick, H. Detmold, A. Cichowski, and R. Hill. Finding camera overlap in large surveillance networks. *ACCV*, 2007.