USING LOCAL REGRESSION KERNELS FOR STATISTICAL OBJECT DETECTION

Hae Jong Seo and Peyman Milanfar

Electrical Engineering Department
University of California at Santa Cruz
1156 High Street, Santa Cruz, CA, 95064

ABSTRACT

We present a novel approach to the problem of detection of visual similarity between a template image, and patches in a given image. The method is based on the computation of a local kernel from the template, which measures the likeness of a pixel to its surroundings. This kernel is then used as a descriptor from which features are extracted and compared against analogous features from the target image. Comparison of the features extracted is carried out using canonical correlations analysis. The overall algorithm yields a scalar resemblance map (RM) which indicates the statistical likelihood of similarity between a given template and all target patches in an image being examined. Performing a statistical test on the resulting RM identifies similar objects with high accuracy and is robust to various challenging conditions such as partial occlusion, and illumination change.

Index Terms— object detection, local metric learning, kernel regression, canonical correlation analysis, test statistic, principal component analysis

1. INTRODUCTION AND OVERVIEW

Analysis of visual objects in images is a very important component in computer vision systems which perform object recognition, image retrieval, image registration, and more. Areas where such systems are deployed are diverse and include such applications as surveillance (security), video forensics, and medical image analysis for computer-aided diagnosis, to mention just a few. The generic problem of interest addressed in this paper can be briefly described as follows: We are given a *single* "template" or "example" image of an object of interest (for instance a picture of a face), and we are interested in detecting similar objects within other "target" images. The target images may contain such similar objects (say other faces) but these will generally appear in completely different context and under different imaging conditions. Examples of such differences can range from rather simple optical or geometric differences (such as occlusion, differing view-points, lighting, and scale changes); to more complex inherent structural differences such as for instance a hand-drawn picture of a face rather than a real face. As an example, we refer the reader to Figure 1 (a). To date, many methods based on such features as histograms, gradients, and shape descriptors have been proposed to address this problem. We refer the interested reader to [1] and references therein for a good summary. Our proposed method is based on the calculation and use of what we call *local regression kernels* which are local weights computed directly from the given pixels in both the template and the target images, as elaborated below. The origin and motivation behind the use of these local kernels is our earlier work on kernel regression for image processing and reconstruction [2]. In that work, we derived localized nonlinear filters which adapt themselves to the underlying structure of the image in order to very effectively perform denoising, interpolation, and even deblurring [3].

The fundamental component of our so-called steering kernel regression method is the calculation of the *local steering kernel* (LSK) which essentially measures the local similarity of a pixel to its neighbors both geometrically and radiometrically. The key idea is to robustly obtain local data structures by analyzing the radiometric (pixel value) differences based on estimated gradients, and use this structure information to determine the shape and size of a canonical kernel. More specifically, the local kernel $K(\cdot)$ is modeled as a radially symmetric function such as a Gaussian.

$$K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x}) = \frac{K(\mathbf{H}_i^{-1}(\mathbf{x}_i - \mathbf{x}))}{\det(\mathbf{H}_i)}, \quad i = 1, \cdots, P^2, \quad (1)$$

where $\mathbf{x}_i = [x_{1i}, x_{2i}]^T$ is the spatial coordinates, P^2 is the number of pixels in a local window and the so-called $steering$ matrix is defined as

$$\mathbf{H}_i^s = h\mathbf{C}_i^{-\frac{1}{2}} \in \mathbb{R}^{(2\times 2)}, \quad (2)$$

where h is the smoothing parameter, and the matrix $\mathbf{C}_i$ is estimated from a collection of spatial gradient vectors $z_{x_1}(\cdot)$ and $z_{x_2}(\cdot)$ within the local analysis window around a sampling position $\mathbf{x}$. The "steering" matrix $\mathbf{H}_i^s$ modifies the shape and size of the local kernel in a way which encodes the local geometric structures present in the image. (See Figure 1 (b) for an example.) We refer the reader to [2] for further details. [1]

This work was supported in part by AFOSR Grant FA9550-07-1-0365

[1] Note that other local kernels such as in [4] and [5] are also applicable in the proposed approach.

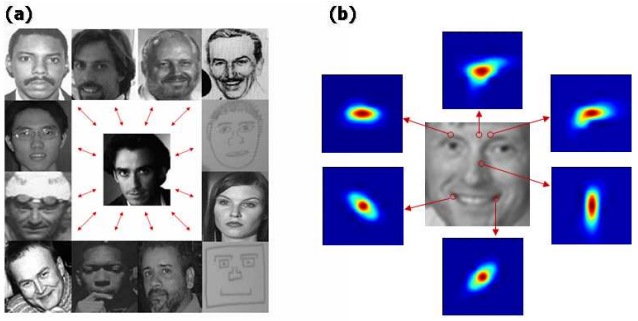

Fig. 1. (a) A face and some possibly similar images (b) Examples of local steering kernel (LSK) in various regions

In what follows, at a position $\mathbf{x}$, we will essentially be using (a normalized version of) the function $K_{\mathbf{H}_i}(\mathbf{x}_i - \mathbf{x})$ as a function of $\mathbf{x}_i$ and $\mathbf{H}_i$ to represent an image's inherent local geometry; and from this function we will extract features which will be used to compare the given patch against patches from another image. As we note later, normalization of this kernel function yields robustness to illumination, contrast, and color differences.

Very recently, a related method by Shechtman et al.[6] introduced a similar matching framework based on the so-called "local self-similarity" descriptor. It is worth mentioning that this (independently derived) local self-similarity measure is a special case of our local steering kernel and is in the same spirit as a number of other local data adaptive metrics which have been used for regression (e.g. [2, 4, 5]) in the image processing community. It is the aim of this paper to begin the process of applying the local kernel idea (in particular the local *steering* kernel) to problems involving detection of similarity across images, and later videos. As mentioned earlier, while the method proposed by Shechtman et al. [6] is related to our method, their approach fundamentally differs from ours in the following respects : 1) Since the calculation of LSK is stable even in the presence of uncertainty in the data [2], our approach is robust even in the presence of noise and even missing pixels; 2) The approach in [6], similar to selective feature techniques such as SIFT [1] filters out "non-informative" descriptors, while in our method we apply Principal Components Analysis (PCA) to the derived LSK in order to learn the most distinctive features of the data; 3) Finally, while [6] explicitly models local and global geometric relationship, we instead apply Canonical Correlations Analysis to the densely derived features which has the desirable property of being affine invariant. From a practical standpoint, it is important to note that the proposed framework operates using a single example of an image of interest to find similar matches; does not require any prior knowledge about the class of objects being sought; and does not require segmentation of the target image. To summarize the operation of the overall algorithm, given an example (i.e. template) patch, we first calculate the LSK [2] from this patch at all pixel locations. Next, a dimensionality reduction step using standard PCA produces a feature vector of modest dimensions. A similar feature vector computed from candidate patches in the target image is then compared against the template feature using Canonical Correlations Analysis (CCA). This last step produces a "resemblance map" showing the likelihood of similarity (i.e. confidence values) between the reference and target patches (See Figures 2 and 4 for a graphical overview.)

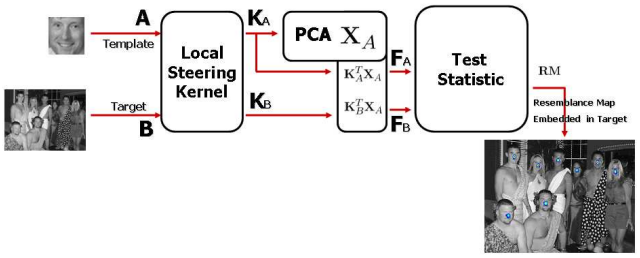

Fig. 2. System overview

The proposed framework is general enough as to be extendable to $3-$D for such applications as action recognition, suspicious behavior detection etc. using an analogous $3-$D local steering kernel [7]. The discussion of this aspect of the ongoing work is outside the scope of this paper. In the next section, we provide further technical details about the various steps outlined above. In Section 3, we demonstrate the performance of the system with some experimental results, and we conclude this paper in Section 4.

2. TECHNICAL DETAILS

Assume that we are given a "target" or test image $B \in \mathbb{R}^{(M\times N)}$ and that we have a template (example) image $A \in \mathbb{R}^{(m\times n)}$ where $M \geq m, N \geq n$. The task at hand is to find patches of B that are similar to A. The first step in the proposed algorithm is to calculate the local steering kernel (LSK) measuring the relationship between a center pixel and its neighborhood pixels, at each pixel from both A and B. As illustrated in the Fig. 1(a), in general, faces are non-rigid and their appearance can vary due to colors, lighting condition, occlusion, rotation,and scale changes, etc. However, the LSK (Fig. 1(b)) captures inherent local geometric properties shared by very different looking faces.

To be more specific, the local steering kernel function $K^j(\mathbf{x}_i - \mathbf{x})$ is calculated and normalized as follows

$$\overline{K}^j_A(\mathbf{x}_i - \mathbf{x}) = \frac{K^j_A(\mathbf{x}_i - \mathbf{x})}{\sum_{i=1}^{P^2} K^j_A(\mathbf{x}_i - \mathbf{x})}\Big|_{j=1,i=1}^{mn,P^2},$$

$$\overline{K}^j_B(\mathbf{x}_i - \mathbf{x}) = \frac{K^j_B(\mathbf{x}_i - \mathbf{x})}{\sum_{i=1}^{P^2} K^j_B(\mathbf{x}_i - \mathbf{x})}\Big|_{j=1,i=1}^{MN,P^2}.$$

As an illustrative detail, we note here that at each reference pixel $\mathbf{x}$, with a preselected window size of $P \times P$, $\overline{K}^j_A(\mathbf{x}_i - \mathbf{x})$ results in an array of P^2 numbers. Therefore, since A is of size $m \times n$, overall $\overline{K}^j_A(\mathbf{x}_i - \mathbf{x})$ is a collection of $P^2 \times mn$ numbers.

To organize these numbers into a manageable array, let $\mathbf{K}_A$,$\mathbf{K}_B$ be matrices whose columns are vectors $\underline{\mathbf{k}}^j_A$,$\underline{\mathbf{k}}^j_B$, which are column-stacked (rasterized) versions of $\overline{K}^j_A$,$\overline{K}^j_B$ respectively, as follows.

$$\mathbf{K}_A = [\underline{\mathbf{k}}^1_A, \cdots, \underline{\mathbf{k}}^{mn}_A] \in \mathbb{R}^{P^2\times mn}, \quad (3)$$

$$\mathbf{K}_B = [\underline{\mathbf{k}}^1_B, \cdots, \underline{\mathbf{k}}^{MN}_B] \in \mathbb{R}^{P^2\times MN}. \quad (4)$$

As described in Fig. 2 the next step is to apply PCA [2] to $\mathbf{K}_A$ for dimensionality reduction and to retain only its salient characteristics. Applying PCA to $\mathbf{K}_A$ we can retain the first (largest) d_x

[2] It is worth noting that the use of the PCA here is not critical in the sense that any dimension reduction method can be used.

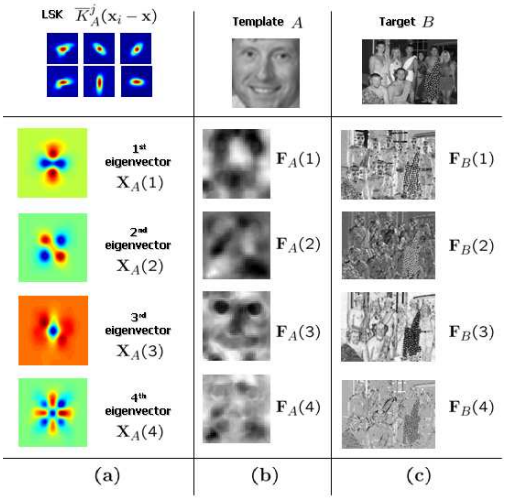

Fig. 3. (a) : $\mathbf{X}_A$ learned from a collection of LSKs $\mathbf{K}_A$, (b): Feature vectors $\mathbf{F}_A$ from template A, (c) : Feature vectors $\mathbf{F}_B$ from target image B. Eigenvectors and feature vectors were transformed to image and up-scaled for illustration purposes.

principal components [3] which form the columns of a matrix $\mathbf{X}_A = [\underline{\mathbf{x}}_{A_1}, \cdots, \underline{\mathbf{x}}_{A_{d_x}}] \in \mathbb{R}^{P^2 \times d_x}$.

Next, the features $\mathbf{F}_A$ and $\mathbf{F}_B$ are obtained by projecting $\mathbf{K}_A$ and $\mathbf{K}_B$ onto the principal subspace defined by $\mathbf{X}_A$ as follows:

$$\begin{aligned} \mathbf{F}_A &= \mathbf{K}_A^T \mathbf{X}_A \in \mathbb{R}^{mn \times d_x}, \\ \mathbf{F}_B &= \mathbf{K}_B^T \mathbf{X}_A \in \mathbb{R}^{MN \times d_x}. \end{aligned} \quad (5)$$

Fig.3 illustrates the principal components in $\mathbf{X}_A$ and shows what the features $\mathbf{F}_A, \mathbf{F}_B$ look like for a particular example.

The next step in the algorithm is the measurement of a "distance" between the computed features, $\mathbf{F}_A$ and $\mathbf{F}_{B(\mathbf{x})}$(=a chunk of $\mathbf{F}_B$ centered around the pixel $\mathbf{x}$). For this purpose we employ the powerful framework of Canonical Correlations Analysis (CCA)[8] as it is pointed out in the previous section. The key idea behind CCA is to find unit direction vectors $\underline{\mathbf{u}}, \underline{\mathbf{v}}$ along which the two sets of variables, $\mathbf{F}_A, \mathbf{F}_{B(\mathbf{x})} \in \mathbb{R}^{mn \times d_x}$ are maximally correlated.

$$\underline{\rho} = \max_{\underline{\mathbf{u}}, \underline{\mathbf{v}}} \frac{(\mathbf{F}_A \underline{\mathbf{u}})^T (\mathbf{F}_{B(\mathbf{x})} \underline{\mathbf{v}})}{\sqrt{(\mathbf{F}_A \underline{\mathbf{u}})^T (\mathbf{F}_A \underline{\mathbf{u}})(\mathbf{F}_{B(\mathbf{x})} \underline{\mathbf{v}})^T (\mathbf{F}_{B(\mathbf{x})} \underline{\mathbf{v}})}}, \quad (6)$$

where $\underline{\mathbf{u}}, \underline{\mathbf{v}}$ are called canonical variates, and $\underline{\rho}$ is canonical correlations and can be computed by a coupled eigenvalue problem based on auto-covariance matrix and cross-covariance matrix of $\mathbf{F}_A$ and $\mathbf{F}_{B(\mathbf{x})}$. The CCA, in addition to maximizing the mutual correlations, has an affine-invariant property which is desirable for similarity detection. We refer the reader to [8] for more detail. Due to the orthogonality of eigenvectors (each column of $\mathbf{X}_A$), the columns of the feature matrix ($\mathbf{F}_A$) are correspondingly mutually uncorrelated. In order to employ CCA, however, the columns of the feature matrix ($\mathbf{F}_A$) should be correlated. To deal with this problem, we vectorize $\mathbf{F}_A$, $\mathbf{F}_{B(\mathbf{x})}$ to $\underline{\mathbf{f}_A}$, $\underline{\mathbf{f}_{B(\mathbf{x})}}$. Then, resulting ρ boils down to a simple correlation coefficient. [4]

[3] d_x is typically selected to be a small integer such as 3 or 4

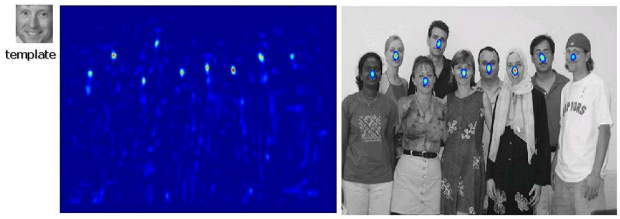

Fig. 4. Left: Resemblance map (**RM**), Right: **RM** embedded in target B after non-maxima suppresion[9].

The final step in the algorithm is to compare the computed canonical correlation value to a threshold [5]which will indicate whether the pair of features from the two images are sufficiently similar or not. If we assume that in the null condition, the local regions of the image are dissimilar, then we would expect the canonical correlation value to be small. But to be more precise, we employ one of the several available significance tests which rely on the assumption of Gaussianity of $\widehat{\rho}$ in the null hypothesis. The larger the value of this test statistic, the more similar the pair of feature vectors are. We use the canonical correlation value to compute a "resemblance map" (**RM**), which will be an image with values indicating the likelihood of similarity between the reference patch and the given image. The value of the resemblance map $\mathbf{RM}(i,j)$ is calculated as follows:

$$\mathbf{RM}(i,j) = \frac{\rho_{(i,j)}^2}{1 - \rho_{(i,j)}^2}, \quad \text{where } \{ \begin{array}{l} i = 1, \cdots, M - m + 1, \\ j = 1, \cdots, N - n + 1. \end{array} \} \quad (7)$$

In Fig.4, an example resemblance map is presented. Red color represents higher resemblance. Here we employ the idea of non-maxima suppression [9] for the final detection. Namely, since the **RM** provides us confidence values, we take the region with the highest value and eliminate the possibility that any other object is detected within some radius of the center of that region again. Then we iterate this process until the next highest value falls below the threshold.

3. EXPERIMENTAL RESULTS

In our experiments, gray-scale images are used as the input (template A and target B). We compute LSK of size $9\times$ 9 as descriptors. As a consequence, each pixel in A and B yields an 81-dimensional local descriptor $\overline{K}$ respectively. Then we reduce this dimensionality by using PCA. We tested our system on some of the MIT+CMU frontal face test set[10]. We used 65 images [6] which contain a total of 233 faces. Fig. 6 shows some results on this dataset. We achieved 81%

[4] We note that this is a special case. More generally, if higher dimensional data such as color images or video are under consideration, we first vectorize the feature matrix from each color channel or each frame and collect them into columns of the feature matrix. In general, these columns of the resulting feature matrix will be correlated and can be treated directly using CCA approach.

[5] For instance, this threshold can be set to achieve a 99 % confidence level computed from an estimate of the empirical pdf of the test statistic.

[6] The current implementation of the algorithm is not robust to large rotations and scale changes; hence we selected a subset of the MIT+CMU database on which to illustrate the results.

detection rate with a handful of false positives. However, for the sake of completeness, we also show the Receiver Operating Characteristic (ROC) curve. A ROC curve representing the performance of our system on this test set is shown in Fig. 5. We also tested our system to detect objects such as airplane, motorbike, car, and ferry on the Caltech 101 dataset [11]. Fig. 7 shows that detection results using one "template" motorbike. The performance of our system appears to be competitive with the method [6] and methods cited therein.

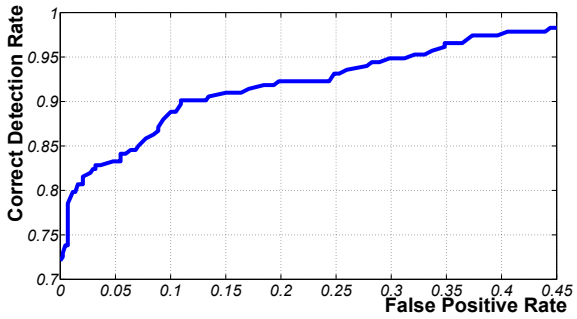

Fig. 5. Receiver Operating Characteristic (ROC) curve on the MIT+CMU test set.

Fig. 6. Detection results on the MIT-CMU dataset[10]. Only one template image is used for matching for all cases.

4. CONCLUSION

In this paper, we have proposed a novel and powerful statistical object detection framework. Our main contribution consists of a general statistical object detection framework based on local steering kernels, and calculation of test statistics derived from canonical correlation analysis. The proposed framework is general enough as to be extendable to 3−D for such applications as action recognition, suspicious behavior detection etc. using analogous 3−D local steering kernel [7]. Improvement of the computational complexity of the proposed method is also a direction of future research worth exploring.

Fig. 7. Detection results on the Caltech 101 object database [11].

5. REFERENCES

[1] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, October 2005.

[2] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, February 2007.

[3] ——, "Deblurring using regularized locally-adaptive kernel regression," *IEEE Transactions on Image Processing*, vol. 17, pp. 550–563, April 2008.

[4] C. Kervrann and J. Bourlanger, "Optimal spatial adapation for patch-based image denoising," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2866–2878, October 2006.

[5] A. Buades, B. Coll, and J. M. Morel, "Nonlocal image and movie denoising," *International Journal of Computer Vision*, vol. 76, no. 2, pp. 123–139, 2008.

[6] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.

[7] H. Takeda, "Nonparametric kernel regression methods for image processing and reconstruction," *A Dissertation Prospectus for Advanced to Ph.D. Candidacy*, Contact author for a copy.

[8] M. Tatsuoka, *Multivariate Analysis*. New York: Macmillan, 1988.

[9] F. Devernay, "A non-maxima suppression method for edge detection with sub-pixel accuracy," *Technical report, INRIA*, no. RR-2724, 1995.

[10] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE PAMI*, vol. 20, pp. 22–38, 2004.

[11] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories." *IEEE. CVPR Workshop on Generative-Model Based Vision*, no. RR-2724, 2004.