Appears in IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, 2005.

Probabilistic Kernels for the Classification of Auto-Regressive Visual Processes

Antoni B. Chan and Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego
abchan@ucsd.edu, nuno@ece.ucsd.edu

Abstract

We present a framework for the classification of visual processes that are best modeled with spatio-temporal autoregressive models. The new framework combines the modeling power of a family of models known as dynamic textures and the generalization guarantees, for classification, of the support vector machine classifier. This combination is achieved by the derivation of a new probabilistic kernel based on the Kullback-Leibler divergence (KL) between Gauss-Markov processes. In particular, we derive the KL-kernel for dynamic textures in both 1) the image space, which describes both the motion and appearance components of the spatio-temporal process, and 2) the hidden state space, which describes the temporal component alone. Together, the two kernels cover a large variety of video classification problems, including the cases where classes can differ in both appearance and motion and the cases where appearance is similar for all classes and only motion is discriminant. Experimental evaluation on two databases shows that the new classifier achieves superior performance over existing solutions.

1. Introduction

Figure 1 presents a sample from a large collection of visual processes that have proven remarkably challenging for traditional motion representations, based on modeling of the individual trajectory of pixels [1, 2], particles [3], or objects in a scene. Since most of the information required for the perception of these processes is contained in the interaction between the many motions that compose them, they require a holistic representation of the associated motion field capable of capturing its variability without the need for segmentation or tracking of individual components. Throughout the years, some representations appeared particularly promising in this respect, e.g. the representation of the motion field as a collection of layers [4]. However, only recently some real success has been demonstrated through the modeling of these processes as dynamic textures, i.e. realizations of an auto-regressive stochastic process with both a spatial and temporal component [5, 6]. Like many other recent advances in vision, the success of these methods derives from the adoption of representations based on generative probabilistic models that can be learned from collections of training examples.

Figure 1. Examples of visual processes that are challenging for traditional spatio-temporal representations: fire and smoke, or the motion of an ensemble of objects, e.g. a school of fish, the traffic on a highway, or the flow of a crowd.

In the context of classification, detection, and recognition problems, the probabilistic representation has various properties that are known to be assets for perception [7], e.g. existence of principled inference formalisms that allow the fusion of diverse sources of information, the ability to incorporate domain-knowledge in the form of prior beliefs, etc. There are, nevertheless, core aspects in which it also has strong shortcomings. In particular, while it can lead to optimal classifiers by simple application of Bayesian decision theory, these classifiers have weak generalization guarantees, and can be quite sensitive to the dimensionality of the underlying feature space, or prone to over-fitting when the models have large numbers of parameters. This is a source of particular concern for the problems of Figure 1 since spatio-temporal autoregressive modeling tends to require high dimensional feature and state spaces.

An alternative classification framework [8], which delivers large-margin classifiers of much better generalization ability (e.g. the now popular support vector machine), does exist but has strong limitations of its own. For the classification of spatio-temporal data-streams, the most restrictive among these is a requirement for the representation of those data-streams as points in Euclidean space. These points are then mapped into a high-dimensional feature space by a kernel function that transforms Euclidean distances in domain space into distances defined over a manifold embedded in range space. The Euclidean representation is particularly troublesome for spatio-temporal processes, where different instances of a process may have different temporal extents (e.g. two similar video streams with different numbers of frames), or be subject to simple transformations that are clearly irrelevant for perception and classification (e.g. a change of sampling rate), but can map the same data-stream into very different points of Euclidean space.

Recent developments in the area of probabilistic kernels have shown significant promise to overcome these limitations. Probabilistic kernels are kernels that act on pairs of generative probabilistic models, enabling simultaneous support for complex statistical inference, which is characteristic of probabilistic representations, and good generalization guarantees, which are characteristic of large-margin learning. Although the feasibility of applying these kernels to vision problems has been demonstrated on relatively simple recognition tasks, e.g. the recognition of objects presented against a neutral background [9], we believe that this is unsatisfactory in two ways. First, the greatest potential for impact of probabilistic kernels is in the solution of classification problems where 1) simple application of Bayesian decision theory is likely to fail, e.g. problems involving large state-space models and high-dimensional features, and 2) the inappropriateness of the Euclidean representation makes the traditional large-margin solutions infeasible. Second, many of the recognition problems for which there are currently no good solutions in the vision literature, e.g. those involving the processes of Figure 1, are exactly of this type.

Both of these points are addressed in this work, which makes contributions at two levels. On one hand, we introduce a procedure for the design of large-margin classifiers for spatio-temporal autoregressive processes. This includes the derivation of a discriminant distance function (the Kullback-Leibler divergence) for this class of processes and its application to the design of probabilistic kernels. On the other, we demonstrate the practical feasibility of large margin classification for vision problems involving complex spatio-temporal visual stimuli, such as the classification of dynamic textures or the classification of patterns of highway traffic flow under variable environmental conditions. The new large-margin solution is shown to perform well above the state of the art and to produce quite promising results for difficult problems, such as monitoring highway congestion.

2. Modeling motion flow

Various representations of a video sequence as a spatio-temporal texture have been proposed in the vision literature over the last decade. Earlier efforts were aimed at the extraction of features that capture both the spatial appearance of a texture and the associated motion flow field. For example, in [10], temporal textures are represented by the first and second order statistics of the normal flow of the video. These types of strictly feature-based representation can be useful for recognition but do not provide a probabilistic model that could be used for kernel design.

More recently, various authors proposed to model a temporal texture as a generative process, resulting in representations that can be used for both synthesis and recognition. One example is the multi-resolution analysis tree method of [11], which represents a temporal texture as the hierarchical multi-scale transform associated with a 3D wavelet. The conditional probability distributions of the wavelet coefficients in the tree are estimated from a collection of training examples and the texture is synthesized by sampling from this model. Another possibility is the spatio-temporal autoregressive (STAR) model of [5], which models the interaction of pixels within a local neighborhood over both space and time. By relying on spatio-temporally localized image features these representations are incapable of abstracting the video into a pair of holistic appearance and motion components.

This problem is addressed by the dynamic texture model of [6], an auto-regressive random process (specifically, a linear dynamical system) that includes a hidden state variable, in addition to the observation variable that determines the appearance component. The motion flow of the video is captured by a dynamic generative model, from which the hidden state vector is drawn. The observation vector is then drawn from a second generative model, conditioned on the state variable. Both the hidden state vector and the observation vector are representative of the entire image, enabling a holistic characterization of the motion for the entire sequence. For this reason, we adopt the dynamic texture model in the remainder of this work.

2.1. The dynamic texture model

The dynamic texture model [6] is defined by

$$x_{t+1} = Ax_t + Bv_t \quad (1)$$

$$y_t = Cx_t + w_t \quad (2)$$

where $x_t \in \mathbb{R}^n$ is a sequence of n-dimensional hidden (state) random variables, $y_t \in \mathbb{R}^m$ a m-dimensional sequence of observed (video frame) random variables, $v_t \sim_{iid} \mathcal{N}(0, I_{n_v})$ a n_v-dimensional driving process (typically $n \ll m$ and $n_v \leq n$), and $w_t \sim_{iid} \mathcal{N}(0, R)$ an observation noise process. The model is parameterized by $\Theta = (A, B, C, R, x_0)$, where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times n_v}, C \in \mathbb{R}^{m\times n}, R \in \mathbb{R}^{m\times m}$, and x_0 is a known initial state. Note that $Bv_t \sim \mathcal{N}(0, Q)$ where $Q = BB^T$. The covariance of the observation noise is assumed to be i.i.d, i.e. $R = \sigma^2 I_m$.

The sequence $\{y_t\}$ encodes the appearance component of the video (video frames), and the motion component is

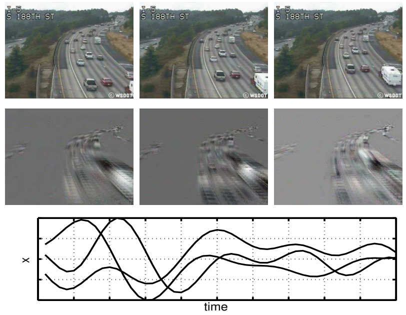

Figure 2. Example of the dynamic texture model: (top) frames from a traffic sequence; (middle) the first three principal components; (bottom) the trajectory of the state space coefficients.

encoded into the state sequence $\{x_t\}$. The hidden state is modeled as a first-order Gauss-Markov process, where the state at time $t+1$, x_{t+1}, is determined by the transition matrix A, the state at time t, x_t, and the driving process v_t. The image at time t, y_t, is a linear combination of the principal components of the entire video sequence, stored in the columns of C, with each component weighted by the corresponding coefficient in the state vector x_t. Figure 2 shows an example of a traffic sequence, its first three principal components, and the state space coefficients.

3. The SVM and probabilistic kernels

A support vector machine (SVM) [8] is a discriminative classifier that constructs a maximum-margin hyperplane between two classes using a set of training examples $\{x_1, \ldots, x_N\} \in \mathcal{X}$. The SVM provides strong generalization guarantees for learning and usually leads to improved performance, outside the training set, when compared to classical methods based on Bayesian decision theory [12]. The training examples that are most difficult to classify are referred to as support vectors, and determine the separating hyperplane.

The SVM can be augmented by using the "kernel" trick, which maps the training examples into a high-dimensional non-linear feature space. This feature space transformation is defined by the kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. One interpretation of the kernel function is that $K(x_i, x_j)$ measures the similarity between the two points x_i and x_j in the space $\mathcal{X}$. A popular example is the Gaussian kernel, defined as $K_g(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$.

If the training examples are represented as probabilistic models (e.g. dynamic textures), the kernel becomes a measure of similarity between probability distributions. A probabilistic kernel is thus defined as a mapping $K : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$, where $\mathcal{P}$ is the space of probability distributions. One such kernel is the Kullback-Leibler kernel [13], defined as $K_{KL}(p, q) = \exp(-\gamma(D(p \,\|q) + D(q \,\|p)))$, where $D(p \,\|q) = \int_{\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} dx$ is the Kullback-Leibler (KL) divergence between the probability distributions $p(x)$ and $q(x)$ [15]. The KL divergence is a natural distance measure between two probability distributions, and the KL kernel in probability space is analogous to the Gaussian kernel in Euclidean space. The KL kernel has been shown to achieve very good results in the domains of object [9] and speech [13] recognition.

4. Probabilistic kernels for dynamic textures

In this section we introduce a probabilistic kernel for visual processes that can be modeled as dynamic textures.

4.1. General considerations

The dynamic texture model provides a probability distribution of the texture in both image and state space. This allows the derivation of two kernels that can ground the classification in either the appearance or the flow of the dynamic texture. Grounding the classification on image space tends to favor iconic pixel matches and leads to best performance when the goal is to differentiate between dynamic textures of different visual appearance (e.g. a flock of birds from a school of fish in Figure 1). Under this approach, two sequences of distinct textures subject to similar motion can be correctly identified. It is, however, not clear that the dynamic texture model is of great significance in this context: a simple appearance classifier based, for example, on the principal component decomposition of the sequences may achieve good results. Ideally, the kernel based on the dynamic texture model should achieve performance at least as good as that of a static kernel, when this is the case.

An alternative classification scenario is that where the different classes have similar appearance and all the discriminant information is contained in the motion flow. For example, problems such as determining the level of traffic on a highway, or detecting outliers and unusual events (e.g. cars speeding or committing other traffic violations). Since for these cases the iconic pixel matching inherent to existing static kernels is clearly inappropriate, these are the cases where dynamic kernels have the greatest potential for improvement over the state of the art.

In summary, depending on the specific classification problem, it may be advisable to ground the classification on either the state space or the image space components of the dynamic texture model. In the remainder of this section we derive the KL kernel for these two representations.

4.2. Probability distributions

We start by obtaining the probability distributions of the Gauss-Markov process that models the state of the dynamic texture. Recursively substituting into (1), it can be

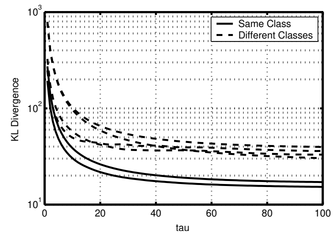

Figure 3. Example of the convergence of the KL divergence rate versus τ. The clustering of generative models in the same and different classes can also be seen.

shown [14] that $x_t \sim \mathcal{N}(\mu_t, S_t)$ where $\mu_t = A^t x_0$ and $S_t = \sum_{i=0}^{t-1} A^i Q (A^i)^T$. Similarly, the covariance between two states x_t and $x_{t'}$ is

$$S_{t,t'} = \begin{cases} A^{t-t'} S_{t'} & , t > t' \\ S^T_{t',t} & , t < t' \end{cases} \tag{3}$$

The image y_t is a linear transformation of the state x_t, and hence $y_t \sim \mathcal{N}(C\mu_t, CS_tC^T + R)$. Furthermore, the covariance between two images y_t and $y_{t'}$ is $\Phi_{t,t'} = CS_{t,t'}C^T$.

4.3. Projection between state spaces

The KL divergence between state spaces cannot be computed directly because each dynamic texture uses a different PCA space. Instead, one state space must be projected into the other by applying a sequence of two transformations: 1) from the original state space into image space, and 2) from image space into the target state space. If the original state space is that of x_1 and the target that of x_2, this is the transformation $\hat{x}_1 = F x_1$ with $F = C_2^T C_1$. From (1),

$$\begin{aligned} x_{t+1} &= A_1 x_t + B_1 v_t && (4) \\ F x_{t+1} &= F A_1 F^{-1} F x_t + F B_1 v_t && (5) \\ \hat{x}_{t+1} &= \hat{A}_1 \hat{x}_t + \hat{B}_1 v_t && (6) \end{aligned}$$

and the transformation of a Gauss-Markov process with parameters (A_1, B_1, x_{01}) is a Gauss-Markov process with parameters $\hat{A}_1 = (C_2^T C_1) A_1 (C_2^T C_1)^{-1}$, $\hat{B}_1 = (C_2^T C_1) B_1$, and $\hat{x}_{01} = (C_2^T C_1) x_{01}$. The KL divergence between state spaces can now be computed with this transformed state model.

4.4. KL divergence between state spaces

The KL divergence rate between two random processes with distributions, $p(X)$ and $q(X)$ over $X = (x_1, x_2, \ldots)$, is defined as

$$D(p(X) \,\|q(X)\,) = \lim_{t\to\infty} \frac{1}{\tau} D(p(x_1^\tau) \,\|q(x_1^\tau)\,)\,. \tag{7}$$

If $p(x_1^\tau)$ and $q(x_1^\tau)$ are the probability distributions of the state sequence $x_1^\tau = (x_1, \cdots, x_\tau)$ of two dynamic textures parameterized by (A_1, Q_1, x_{01}) and (A_2, Q_2, x_{02}), the KL divergence on the RHS of (7) is (see [16] for derivation),

$$\begin{aligned} \frac{1}{\tau} D(p(x_1^\tau) \,\|q(x_1^\tau)\,) = \frac{1}{2} \Bigg[&\log \frac{|Q_2|}{|Q_1|} \qquad (8) \\ &+ \operatorname{tr}(Q_2^{-1} Q_1) - n + \frac{1}{\tau} \|A_1 x_{01} - A_2 x_{02}\|^2_{Q_2} \\ &+ \frac{1}{\tau} \sum_{i=2}^{\tau} \operatorname{tr}\left(\bar{A}^T Q_2^{-1} \bar{A} (S_{i-1} + \mu_{i-1}\mu_{i-1}^T)\right) \Bigg] \end{aligned}$$

where $\bar{A} = A_1 - A_2$, and S_{i-1} and μ_{i-1} are the covariance and mean associated with the state x_{i-1} of the first dynamic texture, and $\|x\|_Q^2 = x^T Q^{-1} x$. In practice, the KL rate can be estimated by setting τ to some large number. Figure 3 shows a graph of some examples of the KL between dynamic textures in state space. The KL rate converges as τ increases, and the clustering between generative models in the same and different classes can be seen.

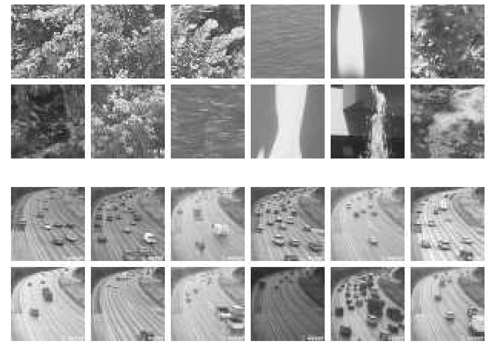

Figure 4. Examples from the databases used for evaluation: (top) the dynamic texture database; (bottom) the traffic video database.

4.5. KL divergence in image space

Since the image probability of a dynamic texture is distributed as a Gaussian, the KL divergence between two image sequence distributions $p(y_1^\tau)$ and $q(y_1^\tau)$ is,

$$\begin{aligned} D(p(y_1^\tau) \,\|q(y_1^\tau)\,) = \frac{1}{2} \Bigg[&\log \frac{|\Phi_2|}{|\Phi_1|} \\ &+ \operatorname{tr}(\Phi_2^{-1}\Phi_1) + \|\gamma_1 - \gamma_2\|^2_{\Phi_2} - m\tau \Bigg] \qquad (9) \end{aligned}$$

where γ_1, Φ_1, γ_2, and Φ_2 are the mean and covariances of the image sequence y_1^τ for the two dynamic textures. Direct computation of the KL divergence between image sequences is intractable since the covariance matrices are $m\tau \times m\tau$, where m is the number of pixels in a frame. Using several matrix identities, it is possible to rewrite the terms of the image KL into a recursive form that is computationally efficient and only requires storing $n\tau \times n\tau$ matrices (recall $n \ll m$). For brevity, we omit the details here and refer the reader to a companion technical report [16].

5. Experimental evaluation

We evaluate the performance of motion flow recognition using the KL kernel on two video databases. The first database contains many visually distinct classes. The second database, based on traffic video, contains visually similar classes, but with varying temporal characteristics.

5.1. Dynamic texture database

The dynamic texture database used in [17] contains 50 classes of various texture, including boiling water, fountains, fire, rippling water, waterfalls, and plants and flowers swaying in the wind. Each class contains four grayscale video sequences with 75 frames of 160×110 pixels. Each sequence was clipped to a 48×48 window that contained the representative motion. Figure 4 (top) shows several examples of the video patches from the dynamic texture database. Since almost all of the classes are visually distinct, the appearance component of the model is likely to be as important for classification as the motion component.

5.2. Traffic video database

The traffic video database consists of 254 video sequences of highway traffic in Seattle, collected from a single stationary traffic camera over two days [18]. The database contains a variety of traffic patterns and weather conditions (e.g. overcast, raining, sunny, rain drops on the camera lens). Each video was recorded in color with a resolution of 320 x 240 pixels with between 42 to 52 frames at 10 fps. Each sequence was converted to grayscale, resized to 80 x 60 pixels, and then clipped to a 48 x 48 window over the area with the most total motion. Finally, for each video clip, the mean image was subtracted and the pixel intensities were normalized to have unit variance. This was done to reduce the impact of the different lighting conditions.

The database was labeled by hand with respect to the amount of traffic congestion in each sequence. In total there were 44 sequences of heavy traffic (slow or stop and go speeds), 45 of medium traffic (reduced speed), and 165 of light traffic (normal speed). Figure 4 (bottom) shows a representative set of clips from this database. All clips are very similar in that the views are obtained with a fixed camera facing the same stretch of road, and the motion is always in the same direction and confined to the same area. Thus, an effective classifier for this problem must be able to distinguish between the different patterns of flow, i.e. the underlying temporal process.

5.3. Experiment setup

The parameters of the dynamic texture model were learned for each video clip using the method of [6]. To ensure that the KL divergence converged, the transition matrix A was scaled so that the largest eigenvalues lie on the unit circle. In addition, the covariance of the driving process was regularized to prevent problems with singular matrices, i.e. we set $Q' = Q + I_n$. All classification results were averaged over four trials. In each trial the data set was split differently with 75% used for training and cross-validation, and 25% reserved for testing.

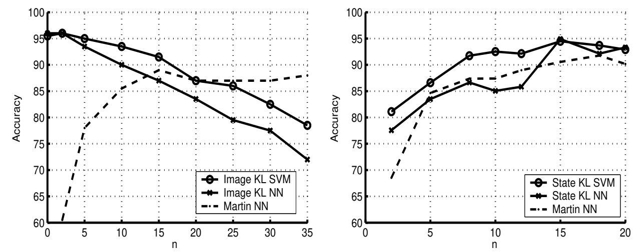

Figure 5. Classification accuracy of the SVM with the KL kernel on two databases: (left) the dynamic texture database using the image KL kernel; (right) the traffic video database using the state KL kernel. The performance of the nearest neighbors classifier using the KL distance and the Martin distance is also shown, and n is the number of principal components used in the dynamic texture.

For the dynamic texture database, SVMs were trained using the KL kernel in image space ($\tau = 25$), and for the traffic video database, the SVMs were trained with the KL kernel in state space ($\tau = 250$). A one-versus-all scheme was used to learn the multi-class problem, and the C and γ parameters were selected using 3-fold cross-validation over the training set. The SVM training and testing was performed using the `libsvm` software package [19]. We also tested a nearest neighbor (NN) classifier using the image space and state space KL as distance measures. Finally, for comparison with the state-of-the-art, a nearest neighbor classifier was implemented using the Martin distance [20,21] as suggested in [17].

5.4. Results

The results on the dynamic texture database, presented in Figure 5 (left), show that the image-based KL classifiers performs significantly better than the Martin distance classifier (an improvement of the best classification accuracy from 89% to 96%). Note that the accuracies of the image KL classifiers improve as the number of principal components n decreases. This was expected since the dynamic texture database contains many visually distinct classes for which the appearance components is more discriminant than the motion. In fact, in the degenerate case of $n = 0$, the video is modeled as a Gaussian whose mean is the mean image of the video sequence, and covariance is the deviation of the frames from this mean. Note how, by achieving top performance for a small number of components, the image-based KL classifiers virtually become static clas-

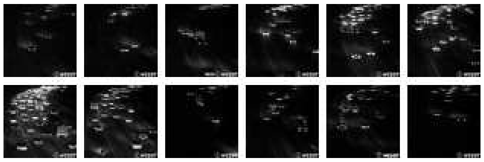

Figure 6. Nighttime traffic videos with a traffic jam.

sifiers. In contrast, the Martin distance nearest neighbors classifier does rather poorly with a small number of components. Hence, although performance improves as n increases, it never reaches an accuracy comparable to that of the KL-based classifiers.

Figure 5 (right) presents the results obtained on the traffic video database. It can be seen from this figure that the two state KL classifiers outperform the Martin NN classifier on this database. Furthermore, all classifiers improve as the number of principal components increases, confirming the fact that a static classifier would do rather poorly on this database. Comparing the performance of the state KL classifiers versus the Martin NN counterpart it can be concluded that 1) the state SVM-KL combination is consistently better, and 2) the state NN-KL combination is better for $n \geq 15$ and also achieves a higher maximum accuracy.

Overall, the image and state KL classifiers outperform the Martin distance nearest neighbor method in classification tasks with both visually distinct video textures, and visually similar, but temporally distinct, video textures. The KL classifiers are also capable of spanning the gamut from static to highly-varying dynamic classifier and, therefore, provide a generic framework for the classification of a large variety of video streams. Comparing the performance of the two KL classifiers, it is clear that SVM-KL combination achieves better classification performance than NN-KL. In particular, the greater robustness of the SVM classifier to a poor selection of the number of components indicates that it has better generalization ability.

Finally, we tested the robustness of the dynamic texture and KL classification framework by using the trained classifiers to label a set of 12 sequential traffic videos that spanned an hour at night, and contained a traffic jam (Figure 6). The NN-KL and SVM-KL correctly labeled the 12 video sequences, including the event of the traffic jam (heavy traffic), and the events of reduced speed (medium) traffic leading up to and immediately following the traffic jam. This is particularly interesting since the classifiers were trained with daytime images containing normally lit cars, yet they are able to correctly label nighttime images where the cars are represented as headlights and a pair of tail lights. These results provide evidence that the dynamic texture model is indeed extracting relevant motion information, and that the proposed classification framework is capable of using the motion model to discriminate between classes of motion. We are currently exploring how this framework could be used for tracking highway congestion, and detection of outlier events such as speeding cars and accidents.

Acknowledgments

We wish to thank Gianfranco Doretto and Stefano Soatto for providing the dynamic texture database used in [17].

References

[1] B. Horn and B. Schunk. Determining Optical Flow. *Artificial Intelligence*, Vol. 17, 1981.

[2] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proc. DARPA Image Understanding Workshop*, 1981.

[3] M. Isard and A. Blake. Condensation – Conditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, Vol. 29(1), pp. 5-28, 1998.

[4] J. Wang and E. Adelson. Representing Moving Images with Layers. *IEEE Trans. on Image Processing*, Vol. 3, Sept. 1994.

[5] M. Szummer and R. Picard. Temporal texture modeling. In *IEEE Conference on Image Processing*, Vol. 3, 1996.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *Intl Journal of Computer Vision*, (2):91–109, 2003.

[7] D. Knill and W. Richards. *Perception as Bayesian Inference*. Cambridge Univ. Press, 1996.

[8] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.

[9] N. Vasconcelos, P. Ho, and P. Moreno. The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition. In *Proc. ECCV*, 2004.

[10] R. Polana and R. C. Nelson. Recognition of motion from temporal texture. In *Proc. CVPR*, pp 129–34, 1992.

[11] Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Trans. on Visualization and Computer Graphics*, 7(2):120–35, 2001.

[12] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[13] P. J. Moreno, P. Ho, and N. Vasconcelos. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *NIPS*, Dec 2003.

[14] S. M. Kay *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993.

[15] S. Kullback. *Information Theory and Statistics*. Dover Publications, New York, 1968.

[16] A. B. Chan and N. Vasconcelos. Efficient Computation of the KL Divergence between Dynamic Textures. Technical Report SVCL-TR-2004-02, `http://www.svcl.ucsd.edu/`, November 2004.

[17] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *Proc. CVPR*, Vol. 2, pp 58-63, 2001.

[18] `http://www.wsdot.wa.gov`

[19] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[20] R. J. Martin. A metric for arma processes. *IEEE Transactions on Signal Processing*, 48(4):1164–70, April 2000.

[21] K. De Cock and B. De Moor. Subspace angles between linear stochastic models. In *IEEE Conference on Decision and Control, Proceedings*, pages 1561–6, December 2000.