SNIPPETS OF SYSTEM IDENTIFICATION IN COMPUTER VISION

Stefano Soatto [*,1] **Alessandro Chiuso** [**,2]

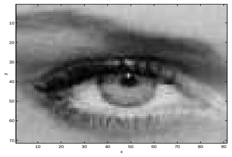
[*] *University of California, Los Angeles – CA 90095,*
soatto@ucla.edu
[**] *Università di Padova – Italy 35131, chiuso@dei.unipd.it*

Abstract: In this paper we illustrate the use of identification-theoretic techniques in computer vision, and hint at some open problems.

Keywords: System identification, computer vision, dynamic scene analysis, visual recognition, dynamic textures, human gaits, dynamic vision.

1. INTRODUCTION

The sense of vision plays a crucial role in the life of primates, allowing them to infer spatial properties of the environment and perform crucial tasks for survival. Primates use vision to explore unfamiliar surroundings, negotiate physical space with one another, detect and recognize a prey at a distance, fetch it, all with seemingly little effort. To this day, engineered systems are far from exhibiting similar capabilities, and endowing a computer with a "sense of vision" is proving to be a formidable task: [3] How can we take a collection of digital images (i.e. arrays of positive numbers) as shown in Table 3, and tell whether we are looking at an apple or a person, and whether she is wearing a hat?



188	186	188	187	168	130	101	99	110	113	112	107	117	140	153	153
189	189	188	181	163	135	109	104	113	113	110	109	117	134	147	152
190	190	188	176	159	139	115	106	114	123	114	111	119	130	141	154
190	188	188	175	158	139	114	103	113	126	112	113	127	133	137	151
191	185	189	177	158	138	110	99	112	119	107	115	137	140	135	144
193	183	178	164	148	134	118	112	119	117	118	106	122	139	140	152
185	181	178	165	149	135	121	116	124	120	122	109	123	139	141	154
175	176	176	163	145	131	120	118	125	123	125	112	124	139	142	155
170	170	172	159	137	123	116	114	119	122	126	113	123	137	141	156
171	171	173	157	131	119	116	113	114	118	125	113	122	135	140	155
174	175	176	156	128	120	121	118	113	112	123	114	122	135	141	155
176	174	174	151	123	119	126	121	112	108	122	115	123	137	143	156
175	169	168	144	117	117	127	122	109	106	122	116	125	139	145	158
179	179	180	155	127	121	118	109	107	113	125	133	130	129	139	153
176	183	181	153	122	115	113	106	105	109	123	132	131	131	140	151
180	181	177	147	115	110	111	107	107	105	120	132	133	133	141	150
181	174	170	141	113	111	115	112	113	105	119	130	132	134	144	153
180	172	168	140	114	114	118	113	112	107	119	128	130	134	146	157
186	176	171	142	114	114	116	110	108	104	116	125	128	134	148	161
185	178	171	138	109	110	114	110	109	97	110	121	127	136	150	160

Table 1. *An "image" (top) is an array of positive numbers (bottom, subsampled) whose values are influenced by many "nuisance factors" including illumination and material properties of the scene.*

This is no easy task, since the measured images depend upon the geometry of the scene (which is unknown), its reflectance properties (also unknown), its motion (unknown) and the illumination of the scene (unknown). While some of these unknowns are not necessarily of interest, they all affect the measurements, and therefore the inference process has to be invariant with respect to these "nuisance" variables. It is easy to convince oneself that from images alone, no matter how many, it is impossible to recover a physically correct model of the geometry (shape), photometry (reflectance) and dynamics (motion) of the scene. In this sense, visual perception *per se* is an ill-

[1] Supported by AFOSR F49620-03-1-0095 and ONR N00014-02-1-0720.

[2] Supported by ASI and MURST.

[3] Although primates appear to be able to "see" effortlessly, it is interesting to notice that about half of their cerebral cortex is devoted to processing visual information (Felleman and van Essen, 1991).

posed problem. However, the inference problem may become well-posed within the context of a specific *task*. For instance, while one cannot infer "the" (physically correct) model of the scene in Figure 2.3, one can infer a *representation* of the scene that can be sufficient to support, for instance, control tasks, or recognition tasks. After all, even if we cannot infer the correct model of steam, or smoke or fire – no matter what computational device we have available, it be a computer or a brain – we sure know how to recognize smoke or fire when we see it.

In this paper we seek to infer models of dynamic visual processes for the purpose of classification and recognition tasks. For instance, from a number of sequences of images of fire, smoke or steam, we want to identify a model that can be used to then recognize, say, fire in a new sequence. The same goes for human motion: how can we infer a model of various gaits, such as walk, run, jump, limp, so that we can for instance detect a limping person from afar? We will first address the simplest possible classes of models, working under the assumption that the underlying data exhibit some form of stationarity. Here the current body of knowledge in system identification has already played a key role in a number of applications, as we describe in Section 2. Even for such simple models, however, recognition and classification tasks remain largely an open problem, which we discuss in Section 3. Imposing simpler models and inferring the model parameters as well as the domain where they are satisfied within a prescribed accuracy leads to a segmentation problem, which is described in Section 4. Extensions to more complex models are countless, and so are their applications. We therefore highlight some open problems in system identification that are motivated by extensions of the research described in this paper in Section 5.

2. MODELING DYNAMIC VISUAL PROCESSES

Since a physically correct model of the *geometry*, *photometry* and *dynamics* of a scene cannot be recovered from visual information alone, it is customary to make *assumptions* on some of the unknowns, in order to infer the others. For instance, in stereo reconstruction one often exploits the assumption that the scene exhibits Lambertian reflection in order to recover shape (see (Ma *et al.*, 2003) and references therein). Naturally, if the assumption is violated, the resulting inference is meaningless, which results in visual illusions that both biological and artificial systems are subject to (Chiuso *et al.*, 2000).

Therefore, here we take a different approach: rather than making prior assumptions and attempt to use visual information to recover a physical model of the scene, we seek to recover a *statistical* model of visual data directly at the outset. Hopefully, this model will support recognition and classification tasks, which we discuss in Section 3. To this end, we represent visual data as the output of a dynamical system driven by realizations of a process drawn from an unknown distribution. Inferring a model of the scene then consists of inferring the model parameters as well as the input distribution. In the next two subsections we illustrate the simplest possible model applied first to pixel intensity, resulting in so-called "dynamic textures", and then to the position of a number of landmark "features", as a model of human gaits.

2.1 Dynamic textures

Let $\{I(t) \in \mathbb{R}^{k\times l}\}_{t=1\dots\tau}$ be a sequence of images. Suppose that at each instant of time t we can measure a noisy version of the image, $y(t) = I(t)+w(t)$ where $w(t)$ is an independent and identically distributed (IID) sequence drawn from a distribution $p_w(\cdot)$ resulting in a positive measured sequence $y(t) \in \mathbb{R}^m,\ t = 1\dots\tau$ [4], where $m = k \times l$. We say that *the sequence* $\{I(t)\}$ *is a (linear) dynamic texture* if there exists a set of n spatial filters $\phi_\alpha,\ \alpha = 1\dots n$ and a stationary distribution $q(\cdot)$ such that, calling $z(t) \doteq \phi(I(t))$, $z(t)$ can be modeled as an ARMA process excited by the white noise $v(t)$, distributed according to $q(\cdot)$. Therefore, a dynamic texture is associated to (a second-order stationary process and, therefore) a state space model with unknown input distribution

$$\begin{cases} x(t+1) = Ax(t) + Bv(t) \\ z(t) = Cx(t) + Dv(t) \\ y(t) = \psi(z(t)) + w(t) \end{cases} \tag{1}$$

with $x(0) = x_0,\ v(t) \overset{IID}{\sim} q(\cdot)$ unknown, $w(t) \overset{IID}{\sim} p_w(\cdot)$ given, and $I(t) = \psi(z(t))$ where $\psi(\phi(I)) = I$. One can obviously extend the definition to an arbitrary non-linear model of the form $x(t+1) = f(x(t), v(t))$, leading to the concept of *non-linear dynamic texture*.

The definition of dynamic texture above, which was proposed in (Doretto *et al.*, 2003), entails a choice of filters $\phi_\alpha,\ \alpha = 1\dots n$. These filters are also inferred as part of the identification process for a given dynamic texture. There are several criteria for choosing a suitable class of filters, ranging from biological motivations to computational efficiency. In the trivial case, we can take ϕ to be the identity, and therefore look at the

[4] This distribution can be inferred from the physics of the imaging device.

dynamics of individual pixels $x(t) = I(t)$ in (1). However, in texture analysis the dimension of the signal is huge (tens of thousands components) and there is a lot of redundancy. Hence we view the choice of filters as a dimensionality reduction step, and seek for a decomposition of the image in the simple (linear) form

$$I(t) = \sum_{i=1}^{n} x_i(t)\theta_i \doteq Cx(t) \tag{2}$$

where $C = [\theta_1, \dots, \theta_n]$ and $\{\theta\}$ can be an orthonormal or overcomplete basis of $\mathcal{L}^2$, a set of principal components, or a wavelet filter bank. In the simples yet effective solution C can be estimated using linear techniques such as principal component analysis. The advantage of this approach, besides simplicity, is that it allows to effectively reduce complexity via a data-tailored construction of basis function. Experimental results show that 20 to 30 principal components yield synthesized textures which are practically indistinguishable from the original ones. Standard approaches based on filter banks require more coefficients to obtain comparable results; for instance, dynamic textures based on Fourier and Gabor filters have been shown in (Zhu and Wang, 2002).

2.2 Human gaits

Instead of modeling the pixel intensities, one could model the position in space of a number of landmark points, for instance the joints of an articulated body. We start from the assumption that a sequence of joint angle trajectories $y(t),\ t = 1 \dots \tau$ is a realization from a second-order stationary stochastic process, i.e.

$$\begin{cases} x(t+1) = Ax(t) + Bv(t) \\ y(t) = Cx(t) + Dv(t) \end{cases} \tag{3}$$

where $v(t)$ is a normalized white noise sequence. Although this may seem like a severely restrictive assumption, we show that it is sufficient to characterize models that are general enough for the purpose of recognition.

2.3 Inference

The problem of going from data to models is the usual system identification problem. Several approaches have been proposed in the literature ranging from standard prediction error methods (Ljung, 1987), to iterative solutions based on expectation-maximization which seem to have obtained a fair amount of attention in the learning community, to the more recent subspace methods (Overschee and Moor, 1993). For the case of dynamic textures, due to the dimension of the signal (76,800 for video at half-resolution), one cannot apply standard identification algorithms. A dimensionality reduction step is a must as we have discussed at the end of section 2.1. Even after this reduction step the signal is high dimensional and therefore subspace methods seems to be best way to go. If we restrict ourselves to first-order AR processes, the following algorithm yields a simple and yet effective solution: Let $Y_1^\tau \doteq [y(1), \dots, y(\tau)] \in \mathbb{R}^{m\times\tau}$ with $\tau > n$, and similarly for X_1^τ and W_1^τ, and notice that

$$Y_1^\tau = CX_1^\tau + W_1^\tau; \quad C \in \mathbb{R}^{m\times n};\ C^TC = I \tag{4}$$

by our assumptions. Now let $Y_1^\tau = U\Sigma V^T; \quad U \in \mathbb{R}^{m\times n};\ U^TU = I;\ V \in \mathbb{R}^{\tau\times n},\ V^TV = I$ be the singular value decomposition (SVD) with $\Sigma = \text{diag}\{\sigma_1, \dots, \sigma_n\}$, and consider the problem of finding the best estimate of C in the sense of Frobenius: $\hat{C}(\tau), \hat{X}(\tau) = \arg\min_{C,X_1^\tau} \|W_1^\tau\|_F$ subject to (4). It follows immediately from the fixed-rank approximation property of the SVD (Golub and Loan, 1989) that the unique solution is given by

$$\hat{C}(\tau) = U \quad \hat{X}(\tau) = \Sigma V^T \tag{5}$$

and $\hat{A}$ can be determined uniquely, again in the sense of Frobenius, by solving the following linear problem: $\hat{A}(\tau) = \arg\min_A \|X_1^\tau - AX_0^{\tau-1}\|_F$ which is trivially done in closed form using an estimate of X from (5):

$$\hat{A}(\tau) = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1} \tag{6}$$

where $D_1 = \begin{bmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{bmatrix}$ and $D_2 = \begin{bmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{bmatrix}$. Notice that $\hat{C}(\tau)$ is uniquely determined up to a change of sign of the components of C and x. Also note that

$$E[\hat{x}(t)\hat{x}^T(t)] \equiv \lim_{\tau\to\infty} \frac{1}{\tau} \sum_{k=1}^{\tau} \hat{x}(t+k)\hat{x}^T(t+k) \simeq \Sigma^2 \tag{7}$$

which is diagonal. Finally, the sample input noise covariance Q can be estimated from

$$\hat{Q}(\tau) = \frac{1}{\tau} \sum_{i=1}^{\tau} \hat{v}(i)\hat{v}^T(i) \tag{8}$$

where $\hat{v}(t) \doteq \hat{x}(t+1) - \hat{A}(\tau)\hat{x}(t)$. In the algorithm above we have assumed that the number of principal components n was given. In practice, this needs to be inferred from the data. In practice this is done from the singular values $\sigma_1, \sigma_2, \dots,$ by choosing n as the cutoff where the value of σ drops below a threshold. A threshold can also be imposed on the difference between adjacent singular values.

Identifying a model from a sequence of 100 frames takes about 5 minutes in MATLAB on a 1GHz pentium® III PC. Synthesis can be performed in

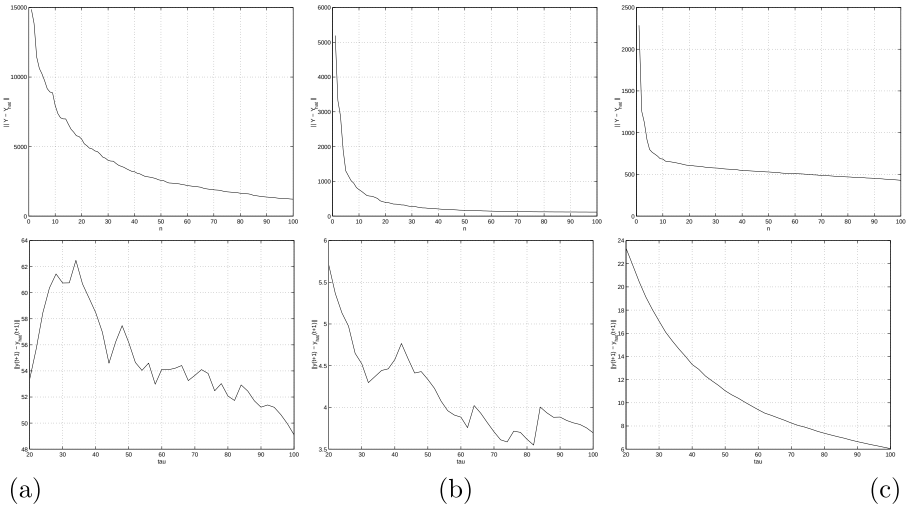


Fig. 1. *Compression error as a function of the dimension of the state space n (top row), and extrapolation error as a function of the length of the training set τ (bottom row). Column (a)* `river` *sequence, column (b)* `smoke` *sequence, column (c)* `toilet` *sequence.*

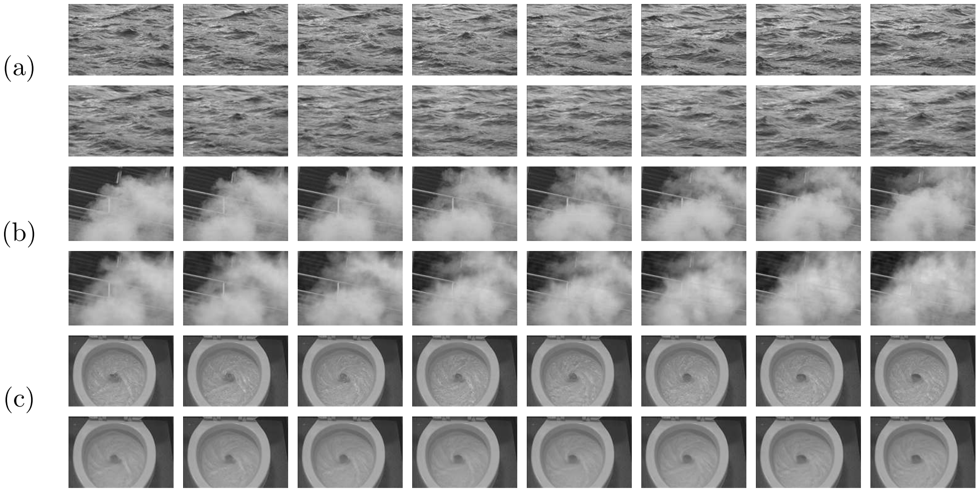


Fig. 2. *(a) river sequence, (b) smoke sequence, (c) toilet sequence. For each of them the top row are samples of the original sequence, the bottom row shows samples of the extrapolated sequence. All the data are available on-line at* `http://www.vision.ucla.edu/projects/dynamic-textures.html`

real time. In our implementation we have used τ between 50 and 150, n between 20 and 50 and k between 10 and 30. Figures 2.3 to 2.3 show the behavior of the algorithm on a representative set of experiments. In each case of Figure 2.3, on the first row we show a few images from the original dataset, on the second row we show a few extrapolated samples. Figure 2.3 shows the overall compression error as a function of the dimension of the state space (top row) as well as the prediction error as a function of the length of the learning set (bottom row). The prediction error is computed by using the first τ images to identify a model, then using the model to predict the image at $\tau + 1$, finally comparing the result with the actual image measured at $\tau + 1$. The plot shows the error between the predicted and the measured images at $\tau + 1$ as a function of τ. A simple histogramming of the residual shows that it deviates considerably from a Gaussian density, with considerable weight at the tails.

3. CLASSIFICATION AND RECOGNITION OF DYNAMIC VISUAL PROCESSES

One approach to classification and recognition is to look directly to the data sequences. These can be regarded as samples from some distribution and therefore classical concepts of discrepancy measures such as Kullback-Leibler divergence, Battacharyya or Hellinger, could be employed. The space of probability distributions, suitably restricted, can be given the structure of a differentiable manifold. One could endow these manifolds with a Riemannian structure, and define distances and probability distributions, which are the basis of standard techniques such as Bayes classification or likelihood ratios. This, however, has several drawbacks. First, one has to take into account that different datasets may have different length. In fact, the probability distribution of an n-vector lives on a different space than that of an m-vector if $m \neq n$. Therefore, it seems more appropriate to look at invariants of the process itself. As we are working with second-order stochastic processes, these are the spectrum, the covariance function or a spectral factor or, in other words, an ARMA model.

ARMA models, learned from data as described in the previous section, do not live on a linear space. The matrix A is constrained to be stable, the matrix C has non-trivial geometric structure after a canonical realization is chosen (for instance, its columns may form an orthogonal set). More in general, the model lives in the quotient space of coordinate transformations of the state space. That leads us to endowing Grassmann and Stiefel manifolds with a metric and probabilistic structure that, although possible, is not straightforward (although see the work of (Hannan and Deistler, n.d.) for references on how to endow transfer functions with the structure of a differentiable manifold.)

While a simple probabilistic approach to classification based on likelihood ratios with respect to a simple probability distribution on Stiefel manifolds has been proposed in (Saisan *et al.*, 2001), a rigorous treatment of this matter has yet to come. In this expository paper, we restrict ourselves to describing an approach to classification and recognition based on the metric structure of the space of models, for instance nearest-neighbor classification or k-means clustering (Duda and Hart, 1973).

Suppose a set of samples $C_1, C_2, \ldots$ is given, where each sample is labelled as belonging to one of c classes λ_j. Given a new sample C, the label λ_m is chosen by taking a vote among the k nearest samples. That is, λ_m is selected if the majority of the k nearest neighbors have label λ_m, which happens with probability

$$\sum_{i=(k+1)/2}^{k} \begin{pmatrix} k \\ i \end{pmatrix} P(\lambda_m|C)^i (1 - P(\lambda_m|C))^{k-i}. \quad (9)$$

It can be shown (Cover and Hart, 1967) that if k is odd the large-sample 2-class error rate is bounded above by the smallest concave function of P^* – the optimal error rate – greater than

$$\sum_{i=0}^{(k-1)/2} \begin{pmatrix} k \\ i \end{pmatrix} \left({P^*}^{i+1} (1 - P^*)^{ki} + {P^*}^{k-i} (1 - P^*)^{i+1} \right) . \quad (10)$$

Note that the analysis holds for k fixed as $n \to \infty$, and that the rule approaches the minimum error rate for $k \to \infty$. For small samples, there are no known results except negative counter-examples that show that an arbitrarily bad error rate can be achieved.

Distances between identified models (which we have inferred using the implementation of the N4SID algorithm (Overschee and Moor, 1993) in the Matlab System Identification Toolbox), can be computed in a number of ways. First, we have computed the "naive" distances (the 2-norm of the difference between corresponding system matrices, without taking the geometry of the manifold into account) and the geodesic distance between models. Not surprisingly these led to quite disappointing results. Then we have computed two metrics between observability subspaces - also taking into account the geometry of the manifold : the Finsler (Weinstein, 1999) and a generalization of the Martin distance, defined in (Martin, 2000) for scalar models. We computed the principal angles between observability subspaces using the algorithm proposed in (Coch and Moor, 2000), where it is formulated for the scalar case but it can be naturally extended to MIMO systems which are innovation models of full-rank vector processes. Then we have calculated the Finsler distance as the maximum subspace angle, and extended the Martin distance d_M by using its relation to the subspace angles θ_i in the scalar case: $d_M = -\ln \prod_i \cos^2 \theta_i$. These two metrics gave similar results, with a slight advantage for the latter. To the distance between learned zero-mean models we added the norm of the difference between the the means of the joint configurations, weighted by a scale factor whose value was set empirically.

For the purpose of illustration, we show the pictorial result of an experiment with three classes of motions that result in similar gaits: walking, running and going up and down a staircase. Notice that these three gaits are quite similar to each other (as opposed, say, to dancing or jumping), and yet the algorithm proposed is capable of correct classification in most cases. These are admittedly preliminary results, and significant work in this area is needed. In Figure 3 we show sample frames from the training datasets. Figure 3 shows the pairwise distance between each model in the dataset. As it can be seen, similar gaits result in smaller distances, with a few outliers. Although this is a very restricted database, it suffices to test our hypothesis.

We have then chosen a few sample sequences for each category as a test sequence. For each of the sequences we have estimated a model by first preprocessing the sequence (after manual initialization) using the ideas described in (Bregler, 1997) to extract joint coordinates, and finally compared the models using a nearest neighbor criterion. A sample frame from the test sequence is shown in

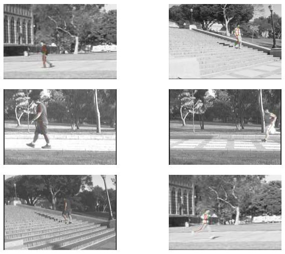


Fig. 3. *Sample frames from the dataset of the gaits: waking, running and walking a staircase.*

Figure 3, while the first two corresponding nearest neighbors are shown to the right. Although this dataset is quite small, the discriminating power of the model as a representation of the dynamic sequence is visible.

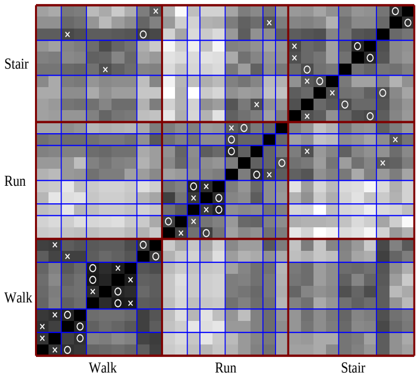


Fig. 4. *The pairwise distance between each sequence in the dataset is displayed in this plot. Each row/column of a matrix represents a sequence, and sequences correspondence to similar gaits are grouped in block rows/columns. Dark indicates a small distance, light a large distance. The minimum distance is of course along the diagonal, and for each row the next closest sequence is indicated by a circle, while the second nearest is indicated by a cross.*

We shall now briefly recall the basics of the distance concept introduced by Martin and elaborated upon by (Coch and Moor, 2000). Let $M_1 \doteq (A_1, C_1, K_1, \Lambda_1)$ and $M_2 \doteq (A_2, C_2, K_2, \Lambda_2)$ be two ARMA models; we define the extended observability matrix

$$\mathcal{O}_\infty(M_i) \doteq \left[C_i^T \;\; A_i^T C_i^T \;\; \ldots \;\; (A_i^T)^n C_i^T \;\; \ldots\right]^T.$$

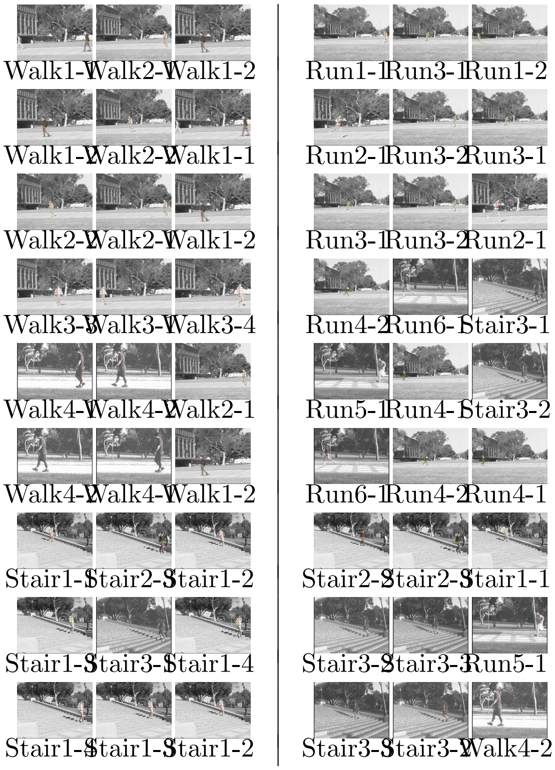


Fig. 5. *For each gait we have chosen a few sample sequences (left) and computed the distance to every other sequence in the dataset. The closest sequence is shown in the central column, while the second nearest is shown in the right column. With a few exceptions, the nearest neighbor belongs to the same gait as the test sequence. Notice that all gaits are quite similar; similar experiments performed on much more diverse gaits such as jumping or dancing return correct classifications.*

It turns out that, as shown in (Coch and Moor, 2000), the concept of distance defined by Martin can be expressed directly in terms of angles between the (extended) observability spaces. In fact, being M_i^{-1} the inverse of the model M_i (recall that M_i is minimum-phase), the distance between M_1 and M_2 can be expressed in terms of the subspace angles between the column spaces of $\left[\mathcal{O}_\infty(M_1) \;\; \mathcal{O}_\infty(M_2^{-1})\right]$ and $\left[\mathcal{O}_\infty(M_2) \;\; \mathcal{O}_\infty(M_1^{-1})\right]$. If we denote by θ_i the i^{th} canonical angle between these spaces the distance defined by Martin is given by

$$d_M(M_1, M_2)^2 = \ln \prod_{i=1}^{2n} \frac{1}{\cos^2\theta_i}, \qquad (11)$$

where n is the model order. Although this is true for scalar ARMA processes, one can measure the distance between multivariable ARMA models using the same concept. In this case the link with the cepstrum is clearly lost. In order to compute subspace angles between models, we proceed as follows

- Compute the solution $Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}$ of the Ljapunov equation

  $$\mathcal{A}^T Q \mathcal{A} - Q = -\mathcal{C}^T \mathcal{C}$$

  where

  $$\mathcal{A} = \begin{pmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_2 - K_2 C_2 & 0 & 0 \\ 0 & 0 & A_2 & 0 \\ 0 & 0 & 0 & A_1 - K_1 C_1 \end{pmatrix}$$

  and

$$\mathcal{C} \doteq \begin{pmatrix} C_1 & -C_2 & C_2 & -C_1 \end{pmatrix}.$$

- Compute the $2n$ largest eigenvalues $\lambda_1, \ldots, \lambda_{2n}$ of
$$\begin{pmatrix} 0 & Q_{11}^{-1}Q_{12} \\ Q_{22}^{-1}Q_{21} & 0 \end{pmatrix}.$$
- $\lambda_i = \cos\theta_i$

We would like to stress that the distance just defined is independent of the basis chosen in the state space. The only requirement is that the model be in innovation canonical form, which is guaranteed by construction in our representation.

It is also worthwhile emphasizing that, as we have anticipated, the innovation variance Λ_i does not enter into the distance computation. This implies that data which differ just by the innovation variance will be treated as deriving from the same model, at least as far as classification/recognition is concerned.

4. SEGMENTATION OF DYNAMIC VISUAL PROCESSES

In modeling the spatio-temporal statistics of a process one is always faced with the conundrum of whether to use a complex model that captures the global statistics, or choose a simple class of models and then partition the scene into regions where the model fits the data within a specified accuracy. In the next section, we discuss a simple model for partitioning the scene into regions where the spatio-temporal statistics, represented by a model identified as we have described above, is constant.

4.1 Spatial segmentation

In this section, which follows (Cremers *et al.*, 2003), we discuss the problem of partitioning a sequence of images into regions that can be described by an ARMA model. Let $\Omega_i \subset \mathbb{R}^2,\ i = 1, \ldots, N$ be a partition of the image into N (unknown) regions [5]. We assume that each pixel contained in the region Ω_i is a Gauss-Markov process. In particular, we assume that there exist (unknown) parameters $A_i \in \mathbb{R}^{n\times n}, C_i \in \mathbb{R}^{m_i \times n}$, covariance matrices $Q_i \in \mathbb{R}^{n\times n}, R_i \in \mathbb{R}^{m_i\times m_i}$, white, zero-mean Gaussian processes $\{v(t)\} \in \mathbb{R}^n,\ \{w(t)\} \in \mathbb{R}^{m_i}$ and a process $\{x(t)\} \in \mathbb{R}^n$ such that the pixels in each region at each instant of time, $y(t)$, obey a model of the type (3). Note that we allow the number of pixels m_i to be different in each region, as long as $\sum_{i=1}^N m_i = m$, the size of the entire image, and that we require that neither the regions nor the parameters change over time, $\Omega_i, A_i, C_i, Q_i, R_i, x_{i,0} = \text{const}$.

Given this generative model, one way to formalize the problem of segmenting a sequence of images is the following: *Given a sequence of images $\{y(t) \in \mathbb{R}^m,\ t = 1, \ldots, T\}$ with two or more distinct regions $\Omega_i,\ i = 1, \ldots, N \geq 2$ that satisfy the model (3), estimate both the regions Ω_i, the unknown state $\{x(t)\}$ in each region and the "signature" of each region, namely the parameters A_i, C_i, the initial state $x_{i,0}$ and the covariance of the driving process Q_i (the covariance R_i is uninformative and therefore excluded from the signature).* Assuming that the parameters A_i, C_i, Q_i, $x_{i,0}$ have been inferred for each region, in order to set the stage for a segmentation procedure, one has to define a discrepancy measure among regions. This has been discussed in the previous section. Now, if the boundaries of each region were known, one could easily estimate a simple model of the spatio-temporal statistics within each region. Unfortunately, in general one does not know the boundaries of each region, and this is one of the goals of the inference process. On the other hand, if the dynamic signature associated with each pixel was known, then one could easily determine the regions by thresholding or by other grouping or segmentation technique. Unfortunately, the model we wish to infer is not a point process, and therefore one cannot pre-compute the signature of each pixel and convert the problem into a static segmentation at the outset. Therefore, we have a classic "chicken-and-egg" problem: If we knew the regions, we could easily identify the dynamical models, and if we knew the dynamical models we could easily segment the regions. Unfortunately, we know neither.

In order to address this problem, one can set up an alternating minimization procedure, starting with an initial guess of the regions, $\hat{\Omega}_i(0)$, estimating the models within each region, $\hat{A}_i(0), \hat{C}_i(0), \hat{Q}_i(0)), \hat{x}_{i,0}$, and then at any given time t seeking for the modification of the regions $\hat{\Omega}_i(t)$, and the update of the models $\hat{A}_i(t), \hat{C}_i(t), \hat{Q}_i(t)), \hat{x}_{i,0}$ so as to minimize a chosen cost functional. For instance, one can minimize the norm of the innovation, integrated in space and time. For the sake of example, in the case of two regions Ω_1, Ω_2, one could seek for [6] $\hat{\Omega}_i(t+1)$ that solves the following optimization problem:

$$\arg\min_{\Omega_i} \sum_{i=1,2} \int_{\Omega_i} \sum_{k=1}^{T} \|y(\xi,k) - \hat{y}_i(\xi,k|k-1)\| d\xi \tag{12}$$

where $y_i(\xi, k|k-1)$ is the predictor of $y(\xi,k)$ based on model i. Once the partition has been estimated one can update $\hat{A}_i(t+1), \hat{C}_i(t+1), \hat{Q}_i(t+1), \hat{x}_0(t+1)$ using for instance subspace identification techniques.

[5] That is, $\Omega = \cup_{i=1}^N \Omega_i$ and $\Omega_i \cap \Omega_j = \emptyset, i \neq j$.

[6] We use the notation $y(\xi,t)$ to indicate the value of a pixel at location $\xi \in \Omega$ at time t.

Under the assumption that each pixel in a region obeys the same dynamical model, the minimum of the corresponding functional is attained when the distance between the two models (one for region Ω_1, the other one for its complement Ω_2) is maximized. Therefore, it is tempting to formulate the problem by simultaneously finding the regions Ω_i and the models M_i by maximizing the distance (11) subject to the constraint that models M_i is identified from data in region Ω_i. A suboptimal approach for this task has been proposed in (Cremers *et al.*, 2003).

For each pixel ξ we generate a local spatio-temporal signature given by the cosines of the subspace angles $\{\theta_j(\xi)\}$ between M_ξ and a reference model, M_{ξ_0}:

$$s(x) = \big(\cos\theta_1(\xi), \dots, \cos\theta_n(\xi)\big). \qquad (13)$$

With the above representation, the problem of dynamic texture segmentation can be formulated as one of grouping regions of similar spatio-temporal signature. We propose to perform this grouping by reverting to the Mumford-Shah functional (Mumford and Shah, 1989). A segmentation of the image plane Ω into a set of pairwise disjoint regions Ω_i of constant signature $s_i \in \mathbb{R}^n$ is obtained by minimizing the cost functional

$$E(\Gamma, \{s_i\}) = \sum_i \int\limits_{\Omega_i} \big(s(\xi) - s_i\big)^2 \, d\xi \, + \, \nu\, |\Gamma|, \quad (14)$$

simultaneously with respect to the region descriptors $\{s_i\}$ modeling the average signature of each region, and with respect to the boundary Γ separating these regions (an appropriate representation of which will be introduced in the next section). The first term in the functional (14) aims at maximizing the homogeneity with respect to the signatures in each region Ω_i, whereas the second term aims at minimizing the length $|\Gamma|$ of the separating boundary.

Let the boundary Γ in (14) be given by the zero level set of a function $\phi : \Omega \to \mathbb{R}$:

$$\Gamma = \{\xi \in \Omega \,|\, \phi(\xi) = 0\}. \qquad (15)$$

With the Heaviside function

$$H(\phi) = \begin{cases} 1 \text{ if } \phi \geq 0 \\ 0 \text{ if } \phi < 0 \end{cases}, \qquad (16)$$

the functional (14) can be replaced by a functional on the level set function ϕ:

$$\begin{aligned} E(\phi, \{s_i\}) = & \int\limits_{\Omega} \big(s(\xi) - s_1\big)^2 H(\phi)\, d\xi \\ & + \int\limits_{\Omega} \big(s(\xi) - s_2\big)^2 \big(1 - H(\phi)\big)\, d\xi \\ & + \nu\, |\Gamma|. \end{aligned} \qquad (17)$$

We minimize the functional (17) by alternating the two fractional steps of:

- Estimating the mean signatures.
  For fixed ϕ, minimization with respect to the region signatures s_1 and s_2 amounts to averaging the signatures over each phase:
  $$s_1 = \frac{\int s\, H(\phi)\, d\xi}{\int H(\phi)\, d\xi}, \quad s_2 = \frac{\int s\, (1 - H(\phi))\, d\xi}{\int (1 - H(\phi))\, d\xi}. \qquad (18)$$
- Boundary evolution.
  For fixed region signatures $\{s_i\}$, minimization with respect to the embedding function ϕ can be implemented by a gradient descent given by:
  $$\frac{\partial \phi}{\partial t} = \delta(\phi) \left[\nu \nabla \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + (s - s_2)^2 - (s - s_1)^2 \right],$$

In Figure 6 we show a few snapshots of the contour evolution, starting from a circle. Notice that the final contour is the contour of an "average" region obtained by combining the different regions in time. Therefore, our approach shows robustness also to changes in the original hypotheses that dynamic textures were spatially stationary.

4.2 Temporal segmentation: filtering and identification of hybrid systems

In addition to partitioning the spatial domain into regions of constant statistics, one can partition the temporal domain, thereby segmenting a continuous process into discrete "events". For instance, the trajectory of joint positions and angles can be partitioned into segments each corresponding to a particular gait, so that we can detect when a walking person begins running or limping.

This is very much related to filtering and identification on hybrid systems and in particular Jump Linear Systems. We have addressed some of the issues related to identifying and filtering linear systems of this kind in (Vidal *et al.*, 2003*a*; Vidal *et al.*, 2002; Vidal *et al.*, 2003*b*) where also one can find a formalization of the problem. We use the techniques developed there to detect spatio-temporal events from live video, for instance the inception of a fire, or an explosion, or the transition from walking to running of a subject in the field of view.

5. EXTENSIONS

All that we have discussed above pertains to systems that are either linear or piecewise linear. However, as we have anticipated, if we identify a model within a linear segment and then plot a histogram of the residual, it is far from Gaussian. An optimality criterion for inference of model and

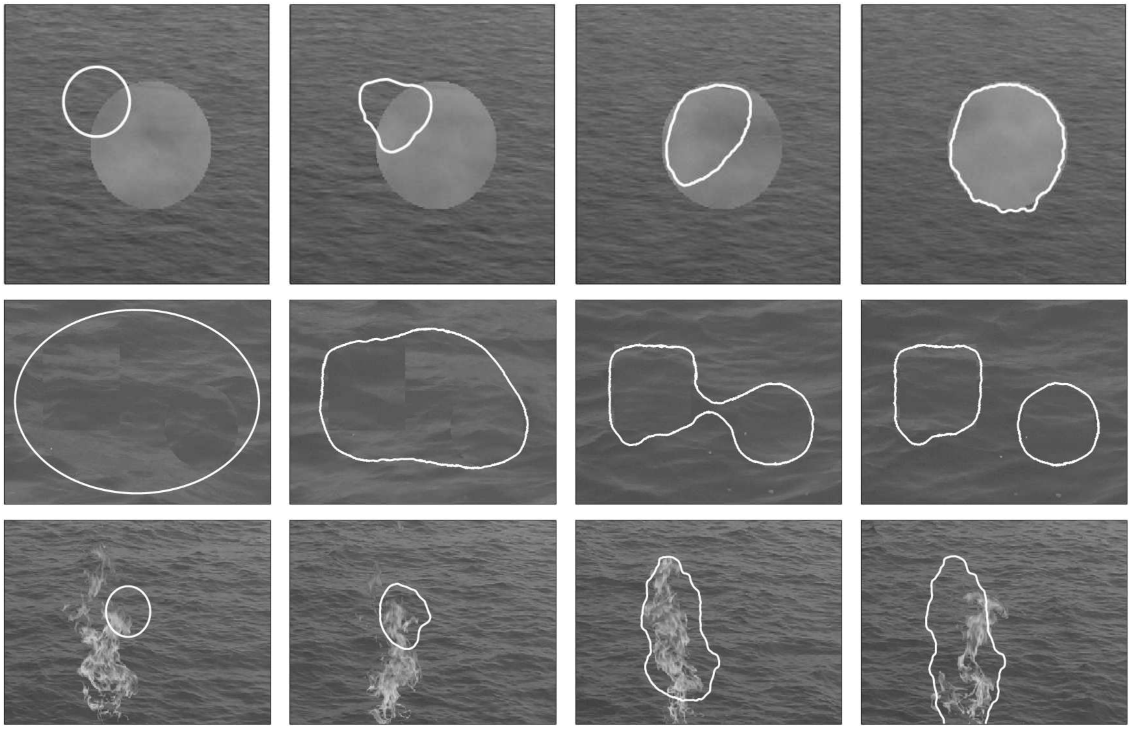


Fig. 6. In this experiment, courtesy of (Cremers et al., 2003), we show the segmentation of scenes based on the dynamic content on different regions. The last example is particularly challenging since the regions change over time. Animation of these results can be downloaded from `http://www.cs.ucla.edu/~doretto/projects/dynamic-segmentation.html`.

input descriptions can be constructed by requiring the estimated input sequence $\hat{v}(t)$ to be a realization from a stochastic process that has maximally independent components. Independence can be expressed in terms of the mutual information among input components, which in turn can be written in terms of the Kullback-Leibler divergence. This approach results in a semi-parametric statistical inference problem, where one has to simultaneously infer the (finite-dimensional) model parameters as well as the (infinite-dimensional) input distribution q. This is essentially an independent component analysis (ICA) problem.

In its conventional static from, ICA attempts to decompose a random vector into a linear combination of statistically independent components. If we call $\mathbf{y} \in \mathbb{R}^m$ the random vector, then ICA looks for a matrix $C \in \mathbb{R}^{m \times n}$ with $n \leq m$ and a random vector $\mathbf{x} \in \mathbb{R}^n$ with independent components, $p_{\mathbf{x}}(x_1, \ldots, x_n) = p_1(x_1) \ldots p_n(x_n)$ such that

$$\mathbf{y} = C\mathbf{x}. \tag{19}$$

The unknowns C and p_i can be estimated by minimizing the mutual information $I(\mathbf{y}||C\mathbf{x}) \doteq \int p_{\mathbf{y}} \log \frac{p_{\mathbf{y}}}{p_{C\mathbf{x}}} d\mathbf{y}$, computed or approximated using a number of independent and identically distributed (IID) samples from $p_{\mathbf{y}}$: $\mathbf{y}(1), \ldots, \mathbf{y}(t) \overset{IID}{\sim} p_{\mathbf{y}}$. Typically the process $\mathbf{y}$ is assumed to be ergodic, and therefore a time series is used in lieu of a fair sample. What we have here, however, is a dynamic ICA problem of separating independent components mixed by linear dynamical (state-space) systems.

Let us rewrite the output of the model at time t:

$$\mathbf{y}(t) = \begin{bmatrix} CA^t, \ CA^{t-1}B, \ldots, CB \end{bmatrix} \begin{bmatrix} \mathbf{x}(0) \\ \mathbf{v}(0) \\ \vdots \\ \mathbf{v}(t-1) \end{bmatrix} \doteq \tilde{\mathcal{C}}^t \tilde{\mathbf{V}} \tag{20}$$

and stack the observations $\mathbf{y}(1), \ldots, \mathbf{y}(t)$ into a vector $\mathbf{Y}^t$ to obtain

$$\mathbf{Y}^t = \tilde{\mathcal{C}}^t \tilde{\mathbf{V}}. \tag{21}$$

One may be tempted to invoke the independence of the components of $\mathbf{V}$ – based on the assumptions that $\mathbf{v}(t)$ is white (time samples are independent) and has independent components – and use standard ICA to estimate the mixing matrix $\tilde{\mathcal{C}}^t$. This, however, does not work because it is not possible to use time realizations as independent samples of $\mathbf{Y}$ due to the initial condition $\mathbf{x}(0)$. One can may conjecture that if t is large enough and A is stable the effect of initial condition will wane; therefore, the assumption may not be as restrictive. Under this assumption, the problem of dynamic ICA can be posed as follows. Consider $\mathbf{Y}^t(k) = [\mathbf{y}((k-1)t)^T, \ldots, \mathbf{y}(kt-1)^T]^T$, and similarly for $\mathbf{V}^t(k)$. Furthermore, let $\mathcal{C}^t$ be the matrix obtaining by completing, in the sense of Toeplitz, the following matrix

$$\begin{bmatrix} CB & & & \\ CAB & CB & & \\ \vdots & \vdots & \ddots & \\ CA^{t-1}B & CA^{t-2}B & \dots & CB \end{bmatrix}. \qquad (22)$$

Then $\hat{A}, \hat{B}, \hat{C}$ can be found sub-optimally by first estimating the mixing matrix $\mathcal{C}^t$ having the particular structure above from a set of independent samples $\mathbf{Y}^t(1), \dots, \mathbf{Y}^t(k)$ (notice that $\mathbf{Y}^t(i)$ and $\mathbf{Y}^t(j)$ do not share components $\mathbf{y}(k)$):

$$\hat{\mathcal{C}}^t(A,B,C) = \arg\min_{\mathcal{C}^t} I(\mathbf{Y}^t(i)||\mathcal{C}^t\mathbf{V}^t(i)) \qquad (23)$$

A suboptimal algorithm for identification based on maximization of input independence has been proposed in (Bissacco and Saisan, 2002). It is based on Amari's natural gradient flow for semiparametric statistical problems (Amari and Cardoso, 1997). Sampling techniques can also be used to perform inference, in a particle filtering framework.

6. DISCUSSION

We have presented a handful of examples where current algorithms for system identification can be successfully employed to address modeling, synthesis and recognition problems in computer vision. These include modeling dynamic textures and human gaits for the purpose of synthesis and classification or recognition. In addition, we have indicated several directions where further work in system identification is needed in order to address difficult tasks of modeling non-Gaussian, non-linear, non-stationary processes for detection, classification, recognition and segmentation.

Acknowledgments

We wish to thank Alessandro Bissacco, Daniel Cremers, Gianfranco Doretto, Paolo Favaro, Payam Saisan, and Ying-Nian Wu for discussions and experimental material used in this paper.

REFERENCES

Amari, S. and F. Cardoso (1997). Blind source separation– semiparametric statistical approach. *IEEE Trans. Signal Processing* **45(11)**, 2692–2700.

Bissacco, A. and P. Saisan (2002). Modaling human gaits with subtleties. In: *Workshop on Dynamic Scene Analysis.*

Bregler, C. (1997). Learning and recognizing human dynamics in video sequences. In: *Proc. of the Conference on Computer Vision and Pattern Recognition.* pp. 568–574.

Chiuso, A., R. Brockett and S. Soatto (2000). Optimal structure from motion: local ambiguities and global estimates. *Intl. J. of Computer Vision* **39**(3), 195–228.

Coch, K. De and B. De Moor (2000). Subspace angles and distances between arma models. *Proc. of the Intl. Symp. of Math. Theory of Networks and Systems.*

Cover, T. and P. Hart (1967). Nearest neighbor pattern classification. *IEEE Trans. on Information Theory* **13**, 21–27.

Cremers, D., G. Doretto, P. Favaro and S. Soatto (2003). Dynamic texture segmentation. In: *UCLA CSD-TR030014.*

Doretto, G., A. Chiuso, Y. Wu and S. Soatto (2003). Dynamic textures. *Intl. J. of Comp. Vis.* **51(2)**, 91–109.

Duda, R. O. and P. E. Hart (1973). *Pattern classification and scene analysis.* Wiley and Sons.

Felleman, D. J. and D. C. van Essen (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex* **1**, 1–47.

Golub, G. and C. Van Loan (1989). *Matrix computations.* 2 ed.. Johns Hopkins University Press.

Hannan, E. J. and M. Deistler (n.d.). *The statistical theory of linear systems.* Wiley and Sons.

Ljung, L. (1987). *System Identification: theory for the user.* Prentice Hall.

Ma, Y., S. Soatto, J. Kosecka and S. Sastry (2003). *An invitation to 3D vision, from images to models.* Springer Verlag.

Martin, R. (2000). A metric for arma processes. *IEEE Trans. on Signal Processing* **48(4)**, 1164–1170.

Mumford, D. and J. Shah (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. on Pure and Applied Mathematics* **42**, 577–685.

Overschee, P. Van and B. De Moor (1993). Subspace algorithms for the stochastic identification problem. *Automatica* **29**, 649–660.

Saisan, P., G. Doretto, Y. Wu and S. Soatto (2001). Dynamic texture recognition. In: *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn..* pp. II 58–63.

Vidal, R., A. Chiuso and S. Soatto (2002). Observability and identifiability of jump-linear systems. In: *Proc. of the Intl. Conf. on Decision and Control.*

Vidal, R., A. Chiuso, S. Soatto and S. Sastry (2003*a*). Observability of linear hybrid systems. In: *Proc. of the Hybrid Systems Computation and Control.*

Vidal, R., S. Soatto and S. Sastry (2003*b*). An algebraic geometric approach to the identification of linear hybrid systems. In: *IEEE Conf. on Decision and Control*. p. (submitted).

Weinstein, A. (1999). Almost invariant submanifolds for compact group actions.

Zhu, S. C. and Y. Z. Wang (2002). A generative method for textured motion: analysis and synthesis. In: *Proc. of the Eur. Conf. on Comp. Vision*.