Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

The Role of Dynamics in Computer Vision and Image Processing

Mario Sznaier Octavia Camps
Electrical and Comp. Engineering Department,
Northeastern University,
Boston, MA 02115.

Abstract— **Dynamic vision and imaging systems can substantially improve our quality of life. However, key issues that must be addressed in order to realize this potential are their fragility when used in unstructured environments and their need to process vast amounts of data in real time. The realization that most actionable information embedded in imaging data can be compactly encapsulated in dynamic models of relatively low order provides a powerful insight to address these issues. As we show in this paper, system theoretic tools allows to recast a wide range of dynamic vision problems into a computationally tractable optimization form. These ideas are illustrated with several applications including tracking, segmentation, and texture analysis/synthesis.**

I. Introduction

Dynamic vision and imaging – the confluence of dynamics, computer vision, image processing and control – is uniquely positioned to enhance the quality of life for large segments of the general public. Aware sensors endowed with tracking and scene analysis capabilities can prevent crime and reduce time response to emergency scenes. Enhanced imaging methods can substantially reduce the amount of radiation required in medical procedures. Moreover, the investment required to accomplish these goals is relatively modest, since a large number of imaging sensors are already deployed and networked. The challenge now is to develop a theoretical framework that allows for *robustly* processing this vast amount of data, within the constraints imposed by the need for real time operation in dynamic, partially stochastic scenarios.

Actionable information buried within imaging data can often be compactly encapsulated in dynamic models that have far lower rank than the dimensionality of the original data. Indeed, the goal of this paper is to illustrate the central role that dynamic models and their associated predictions can play in developing a comprehensive, computationally tractable robust dynamic vision and imaging framework. Establishing a connection with a rich set of robust systems theory tools, in particular interpolation tools developed in the context of robust identification and model (in) validation, allows for recasting a wide spectrum of problems into a tractable, finite dimensional convex optimization. Furthermore, *in many cases merely postulating an underlying model leads to efficient solutions* that rely in dynamical systems tools to detect similarities or differences in model properties, without explicitly finding these models, which is typically a far more demanding task. In turn, computer vision and image processing can provide a rich environment both to draw inspiration from and to test new developments in systems theory. For instance, the applications addressed in this paper point out to the need for further research into low complexity nonlinear identification methods, worst-case identification methods for switched systems, and robust identification/(in)validation of 2-D systems.

II. Notation

$\mathcal{H}_{\infty,\rho}$ space of functions analytic in $|z| \leq \rho$, equipped with the norm $\|G\|_{\infty,\rho} \doteq \sup_{|z|<\rho} \overline{\sigma}\left(G(z)\right)$, where $\overline{\sigma}\left(.\right)$ denotes maximum singular value.

$\mathcal{BH}_\infty(K)$open K–ball in $\mathcal{H}_\infty$

III. Multiframe Tracking

A requirement common to most dynamic vision applications is the *ability to track* objects in a sequence of frames. Current approaches integrate correspondences between individual frames over time, through a combination of some assumed simple target dynamics (e.g. constant velocity), empirically learned noise distributions and past position observations [20], [31]. However, while successful in many scenarios, these approaches still remain vulnerable to model uncertainty, occlusion and appearance changes, as illustrated in Figure 1.

As shown next, this difficulty can be solved by modeling the motion of the target as the output of a dynamical system, to be identified from the available data. To this effect, start by modeling y_k, the position of a given target feature as:

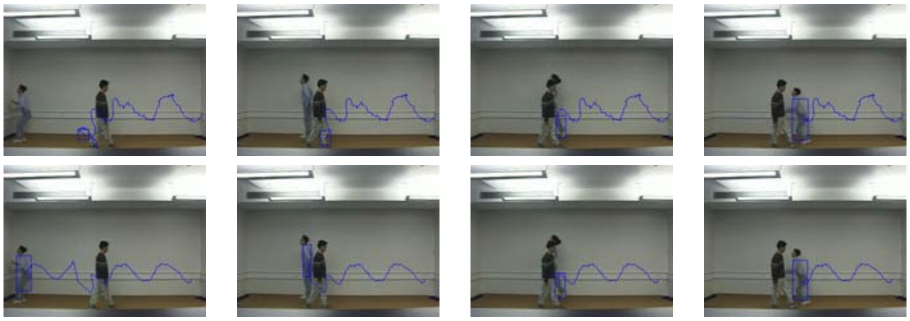
$$y(z) = \mathcal{F}(z)e(z) + \eta(z) \tag{1}$$

where $\mathbf{e}$ and $\eta_k \in \mathcal{N}$ represent a suitable input and measurement noise, respectively. Further, we will assume that the following *a priori* information is available:

(a) Set membership descriptions $\eta_k \in \mathcal{N}$ and $e_k \in \mathcal{E}$. These can be used to provide deterministic models of the stochastic signals e, η.

(b) $\mathcal{F}$ admits an expansion of the form $\mathcal{F} = \overbrace{\sum_{j=1}^{N_p} p_j \mathcal{F}^j}^{\mathcal{F}_p} + \mathcal{F}_{np}$. Here $\mathcal{F}^j$ are known, given, not necessarily stable operators that contain all the information available about possible modes of motion of the target.

This work was supported in part by NSF grants ECS–0648054, IIS–0713003, and AFOSR grant FA9550–05–1–0437.



Frame 150 Frame 105 Frame 95 Frame 85

Fig. 1. Tracking in the presence of occlusion. Top: Unscented Particle Filter based tracker. Bottom: Combination Identified Dynamics/Kalman Filter.

(c) $\mathcal{F}_{np} \in \mathcal{BH}_{\infty,\rho}(K)$ for some known $\rho \leq 1$, e.g. a bound on the divergence rate of the approximation error of the expansion $\mathcal{F}_p$ to $\mathcal{F}$ is available.

In this context, the next location of the target feature y_k can be predicted by first identifying the relevant dynamics $\mathcal{F}$ and then using it to propagate its past values. In turn, identifying the dynamics entails finding an operator $\mathcal{F}(z) \in \mathcal{S} \doteq \{\mathcal{F}(z) \colon \mathcal{F} = \mathcal{F}_p + \mathcal{F}_{np}\}$ such that $y - \eta = \mathcal{F}e$, precisely the class of interpolation problem addressed in [34]. As shown there, such an operator exists if and only if the following set of equations in $\mathbf{p}, \mathbf{h}$ and K is feasible:

$$\mathsf{M}_R(\mathbf{h}) = \begin{bmatrix} \mathsf{R}_\rho^2 & \mathsf{T}_h^T \\ \mathsf{T}_h & K^2 \mathsf{R}_\rho^{-2} \end{bmatrix} \geq 0 \qquad (2)$$

$$\mathbf{y} - \mathsf{T}_u \mathsf{P} \mathbf{p} - \mathsf{T}_u \mathbf{h} \in \mathcal{N} \qquad (3)$$

where T_x denotes the Toeplitz matrix associated with a sequence $\mathbf{x} = [x_1, \ldots, x_n]$, $\mathsf{R}_\rho \doteq \mathrm{diag}\,[1\ \rho\ \cdots\ \rho^n]$, $\mathsf{P} \doteq [f^1\ f^2\ \cdots\ f^{N_p}]$,where f^i is a vector containing the first n Markov parameters of the transfer function $\mathcal{F}^i(z)$ and $\mathbf{h}$ contains the first n Markov parameters of $\mathcal{F}_{np}(z)$.

A Simple Tracking Example: Consider again the problem illustrated in Figure 1. The experimental information consists of centroid position measurements from the first 20 frames, where the target is not occluded. The *a priori* information, estimated from the non–occluded portion of the trajectory is:

1) 5% noise level
2) $\mathcal{E} = \delta(0)$, i.e. motion of the target was modelled as the impulse response of the unknown operator F^1.
3) $\mathcal{F}_p \in \mathrm{span}[\frac{1}{z-1}, \frac{z}{z-a}, \frac{z}{(z-1)^2}, \frac{z^2}{(z-1)^2}, \frac{z^2 - \cos\omega z}{z^2 - 2\cos\omega z + 1}, \frac{\sin\omega z^2}{z^2 - 2\cos\omega z + 1}]$ where $a \in \{0.9, 1, 1.2, 1.3, 2\}$ and $\omega \in \{0.2, 0.45\}$
4) $F_{np} \in \mathcal{BH}_{\infty,\rho}(K)$, with $\rho = 0.99$

As shown in Figure 1, a Kalman filter tracker that uses the identified dynamics is now able to track the target past the occlusion. It is worth emphasizing that this combination significantly outperforms a tracker based solely on an unscented particle filter [20]. Hence, exploiting dynamical information through the use of control–motivated tools, leads to *both* robustness improvement and substantial computational complexity reduction. In addition, the framework described above furnishes *deterministic, worst–case bounds* on the prediction error that can be used to disambiguate among targets with neighboring tracks. Let $\mathcal{T}(\mathbf{y}) \doteq \{\mathcal{F} \in \mathcal{S} \colon y_{k+1} = \mathcal{F}(z)e(z) + \eta(z),\ \eta_k \in \mathcal{N}\}$ be the *consistency set* – i.e. the set of all models consistent with both the *a priori* information and the experimental data. Since the identification procedure used here is *interpolatory*, the generated model $\mathcal{F}_{id}$ belongs to the consistency set $\mathcal{T}(\mathbf{y})$ and its worst case prediction error is given by:

$$\|\hat{\mathbf{y}} - \mathbf{y}\|_* \leq \sup_{\mathcal{F}_1, \mathcal{F}_2 \in \mathcal{T}(\mathbf{y})} \|\mathcal{F}_1[\mathbf{y}, \mathbf{e}] - \mathcal{F}_2[\mathbf{y}, \mathbf{e}]\|_* = \mathcal{D}[\mathcal{T}(\mathbf{y})] \qquad (4)$$

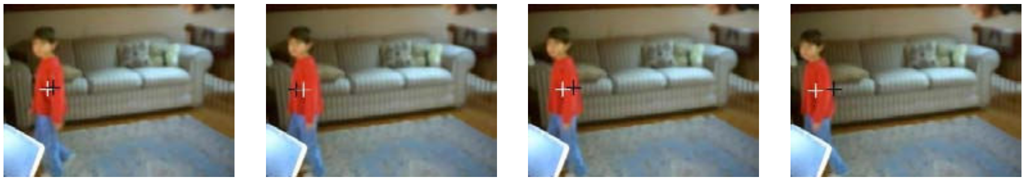
where $\|.\|_*$ is a suitable norm and $\mathcal{D}(.)$ denotes the diameter of the set $\mathcal{T}(\mathbf{y})$. When the sets $\mathcal{S}$ and $\mathcal{N}$ are convex, computing this bound reduces to a convex optimization [39, Lemma 10.3]. Note that these bounds are computed only once and remain valid as long as the underlying dynamics do not change.

Fig. 2 compares the actual and upper bound of the error in a child tracking application. In this experiment the measured position in frame 12 was propagated forward using the identified dynamics and the bounds computed by solving a single Linear Programming problem. If other targets with similar dynamics or photometric properties are present, trackers can safely discard candidates falling outside these bounds. In addition, these bounds provide a mechanism to balance computational requirements and data obsolescence. For instance, in this example these bounds establish *a priori* that no new data is required from Frame 12 until Frame 20, where the error becomes comparable with the width of the target: 30 pixels.

A. Dynamic Appearance Modeling.

Arguably, one of the hardest challenges in tracking is to overcome changes to the target appearance due to articulations, illumination changes, etc. In principle, this difficulty can be solved by using *dynamic* appearance models obtained using the same robust identification approaches employed to identify the motion dynamics [23]. However, moving beyond a few simple descriptors requires addressing the issues of

[1] This is equivalent to lumping together the dynamics of the plant and the input signal.



Frame	13	14	15	16	17	18	19	20
Mean–Shift	25.90	35.93	41.32	45.63	54.65	57.53	65.05	64.80
Actual error	8.87	6.14	10.04	13.03	10.31	15.72	19.50	26.04
Worst case bound	13.00	15	17	19	21	23	25	27

Fig. 2. Top: Prediction (black cross) versus Ground Truth (white cross). Bottom: Id error.

high computational costs, due to the poor scaling properties of LMI based identification algorithms.

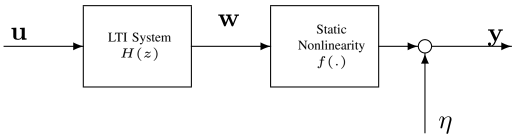


Fig. 3. Wiener System Structure

This challenge can be addressed through the use of nonlinear dimensionality reduction techniques to map the data to a lower dimensional manifold where the identification/tracking is performed. Since the projection onto the lower dimensional manifold can be modeled as a static nonlinearity, this approach leads naturally to a Wiener system structure of the form illustrated in Figure 3, consisting of the interconnection of a LTI system $H(z)$ and a memoryless nonlinearity $f(.)$. Next, we illustrate the effectiveness of this approach using the problem of human motion modeling and tracking. The experimental data, partially shown in Figure 5(a) consists of the first 20 frames of a human walking, each having 1728 pixels. Thus, modeling pixel evolution become infeasible even when using just a few frames. On the other hand using the risk–adjusted approach proposed in [27] and the following *a priori* information

1.- $\omega \in R^3$ (since it represents the coordinates of the centroid of the target).

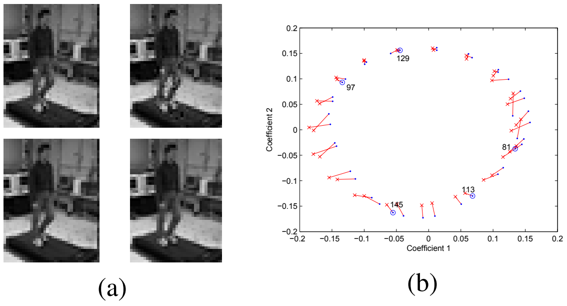
2.- The static nonlinearity $\mathbf{f}(.)$ has the form[2]: $\mathbf{f}(\mathbf{x}) = \mathbf{B}\mathbf{\Psi}(\mathbf{x})$ where $\mathbf{B} \in R^{1726\times 6}$ is an unknown matrix and the bases $\mathbf{\Psi}(\mathbf{x})\colon R^3 \rightarrow R^6$ are given by:

$$\mathbf{\Psi}(\mathbf{x}) = [\exp(-0.8\|\mathbf{x}-\mathbf{t_1}\|_2^2), \\ \exp(-0.8\|\mathbf{x}-\mathbf{t_2}\|_2^2), 1, \mathbf{x}^T]^T$$

where

$$\mathbf{t_1} = \begin{bmatrix}0.6833 & -0.4521 & -0.0033\end{bmatrix}$$
$$\mathbf{t_2} = \begin{bmatrix}-0.7552 & 0.4997 & 0.0036\end{bmatrix}$$

led to a model with a fifth order linear portion that interpolates the data within 10%. In addition, as shown in Figure 5(b), the temporal evolution of the points on the manifold closely agree with the predictions of the linear dynamic model. This substantiates the conjecture posed in [29], that human motion tracking can be decoupled into: (a) a linear tracking problem in a low dimensional manifold, accounting for the *dynamics* of the motion, and (b) a nonlinear, static mapping that accounts for the changes in appearance of the target. Fig. 4 illustrates the application of these ideas to tracking multiple people in an outdoor scene.



(a) (b)

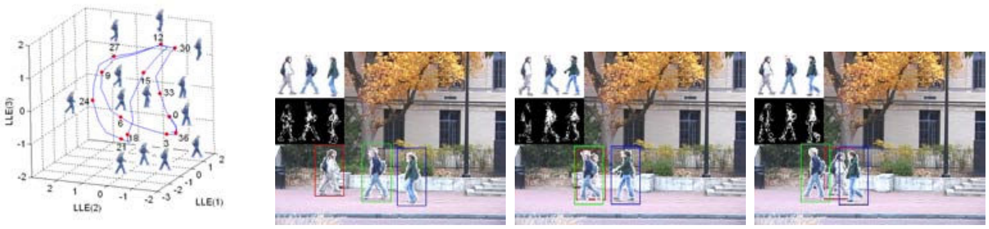
Fig. 5. Learning appearance using a Wiener system. (a) Top: Walking sequence (from CMU MoBo database), Bottom: impulse response of the identified Wiener system. (b) Evolution on a 2D projection of the 3D manifold: predicted (cross) and actual (dot).

B. Receding Horizon Rank Minimization Based Tracking

An implicit assumption in the methods described above is that *the dynamics of interest are linear and do not change,* e.g. the underlying model is linear time invariant (LTI) which allows to identify first and then propagate the dynamics using a standard filter. Extending this approach to the case of slowly varying dynamics requires an on-line implementation–either re-identifying the plant at each instant or performing on-line model (in)validation and re-identifying only when necessary–which could be problematic given the relatively high computational complexity entailed in both processes. Moreover, in addition to divergence problems that could arise from errors in estimating the dynamics, Kalman-filtered based approaches can fail (e.g. lead to unbounded error covariance) in the presence of intermittent observations [40]. This effect can be mitigated by resorting to a Receding Horizon based approach [46], but this further increases the computational complexity and does not address divergence issues due to miss-identified dynamics.

These difficulties can be avoided by using a simple approach [6] inspired on earlier work on subspace identification

[2] This hypothesis is motivated by the bases proposed in [12] to model human silhouettes.



(a) (b)

Fig. 4. (a) Sample 3 dimensional manifold extracted from the sequence shown in (b) and use of dynamics on this manifold to predict target position and appearance.

[28], [33], rank-minimization based track matching [8] and receding horizon based estimation ([24], [22], [15], [1] (and references therein), where trajectories of piece-wise linear plants are interpolated/extrapolated without identifying the dynamics of the plant. Intuitively, the idea is to add new data to the available measurements in a way such that the extended or completed trajectory (for example in the presence of occlusion) can be explained by the same model explaining the available measurements alone. Recalling that the rank of the Hankel matrix is an estimate of the order of the underlying dynamical system, this approach allows to recast the problem into a rank-minimization form [9] that estimates unknown data points by minimizing on-line the rank of a Hankel matrix constructed using the available data and the unknown new data. While rank minimization problems are NP hard, the estimates can be obtained by using a convex relaxation proposed in [14].

More formally, consider (unknown) single input single output linear shift invariant plants with McMillan degree n:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathsf{A}\mathbf{x}_k + \mathsf{B}u_k \\ \zeta_k &= \mathsf{C}\mathbf{x}_k \\ y_k &= \mathsf{C}\mathbf{x}_k + v_k \end{aligned} \tag{5}$$

where $\mathbf{x} \in R^n$, u, ζ and y represent the states, inputs, outputs and measurements corrupted by noise v, respectively, and where the realization (A, B, C) is minimal. Alternatively, it can also be represented by its transfer function:

$$\begin{aligned} \zeta(z) &= G(z)u(z) \\ G(z) &\doteq \frac{\sum_{i=0}^{n} b_i z^{-i}}{1 + \sum_{i=1}^{n} a_i z^{-i}} \end{aligned} \tag{6}$$

Then, the approach for rank minimization tracking is summarized in the two algorithms below:

Algorithm 1: RANK MINIMIZATION BASED TRACKING

Input: n_f, number of features being tracked; the measurements matrix $\mathcal{W} \in R^{2n_f \times N_w}$, where $w_{i,k} = u^i_k$ and $w_{i+1,k} = v^i_k$ are the i^{th} feature position in the k^{th} frame; length of the observation window, N_w; prediction horizon N_p; noise bound ϵ.
Output: Estimated target location $w_{i,k}$ at time $\{k \geq t\}$

```
1.  While {tracking continues} {
        for all $i \in \{1, \cdots, 2n_f\}$ do
            Apply Algorithm 2 on $\{w_{i,k}\}_{k=t-N_w}^{t-1}$
            to compute $\{{w^*}_{i,k}\}_{k=t}^{t+N_p-1}$.
        end for
2.      Locate target around the predicted position and
        update $\{w_{i,t}\}$, otherwise use the value $\{{w^*}_{i,t}\}$
        instead (target is occluded).
3.  t=t+1
    }
```

Algorithm 2: RECEDING HORIZON RANK MINIMIZATION BASED PREDICTION/INTERPOLATION

Input at time k: N_h: Horizon length; $\mathcal{I}_a \subseteq [k - N_h, k]$, (with $\text{card}(\mathcal{I}_a) \geq n$): set of indices of available measurements; $\mathcal{I}_e \subseteq [k - N_h, k+1]$: set of indices of data to be estimated; with $\mathcal{I}_a \cup \mathcal{I}_e = \mathcal{I}$; input/output data $y_\ell,\ \ell \in \mathcal{I}_a$, $u_\ell,\ \ell \in \mathcal{I}$;
set membership description of the measurement noise $v \in \mathcal{N}$.
Output: Estimates $\hat{\zeta}_\ell$ of ζ_ℓ, $\forall \ell \in \mathcal{I}_e \cup \mathcal{I}_a$

1. Let ζ^* denote the following sequence: $\zeta^*_i = \begin{cases} y_i - v_i \text{ if } i \in \mathcal{I}_a \\ x_i \text{ if } i \in \mathcal{I}_e \end{cases}$ where v, x are free variables, and form the matrix $\mathsf{H}(x, v) \doteq \begin{bmatrix} \mathsf{H}_\zeta \\ \mathsf{H}_u \end{bmatrix}$ where

$$\mathsf{H}_\zeta \doteq \begin{bmatrix} \zeta^*_{i_1} & \zeta^*_{i_2} & \cdots & \zeta^*_{i_{n+n_u+1}} \\ \zeta^*_{i_2} & \zeta^*_{i_3} & \cdots & \zeta^*_{i_{n+n_u+2}} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta^*_{i_{n+1}} & \zeta^*_{i_{n+2}} & \cdots & \zeta^*_{i_{2n+n_u+1}} \end{bmatrix}$$

$$\mathsf{H}_u \doteq \begin{bmatrix} u_{i_1} & u_{i_2} & \cdots & u_{i_{n+n_u+1}} \\ u_{i_2} & u_{i_3} & \cdots & u_{i_{n+n_u+2}} \\ \vdots & \vdots & \ddots & \vdots \\ u_{i_{n_u}} & u_{i_{n_u+1}} & \cdots & u_{i_{n+2n_u+1}} \end{bmatrix}$$

2. (approximately) minimize rank$[\mathsf{H}(x,v)]$ by solving the following convex problem in x, v, R, S:
   minimize $Tr(\mathsf{R}) + Tr(\mathsf{S})$
   subject to $\begin{bmatrix} \mathsf{R} & \mathsf{H}(x) \\ \mathsf{H}(x)^T & \mathsf{S} \end{bmatrix} \geq 0$
   subject to: $\{v_\ell\} \in \mathcal{N}$.
3. Estimate/predict the output ζ_ℓ from the noisy measurements y_ℓ by:
   $\hat{\zeta}_i = \begin{cases} y_i - v_i \text{ if } i \in \mathcal{I}_a \text{ (estimation)} \\ x_i \text{ if } i \in \mathcal{I}_e \text{ (interpolation/prediction)} \end{cases}$

The benefits of this approach are illustrated by the following example where tracking using a receding horizon rank minimization filter (RHRMF) is compared against tracking results obtained using a Kalman filter and a combination identification via Caratheodory-Fejer/ Particle Filter (CF-PF) [2]. In this example, the goal is to track the two individuals shown in Figure 6 through occlusion, using video-data obtained with a moving camera. In this case, as standard in the field, the Kalman filter used an assumed simple model of the dynamics, in this case constant velocity, together with the observed data, to estimate and propagate the states and estimate the positions during occlusion. The CF-PF combination used the unoccluded data to identify first the dynamics of the target, followed by the use of these dynamics in conjunction with a particle filter [21] to estimate the target position during occlusion. Finally, the receding horizon rank minimization filter was implemented using *Algorithms* 1 and 2 with the values $N_w = 35$, $N_p = 6$, and $\epsilon = 2$. The results after processing are shown in the bottom portion of Figure 6. As shown there, the receding horizon filter yields the lowest prediction error. This is due to the fact that the simple model used in the Kalman filter does no completely capture the target dynamics. These dynamics are captured by the CF-based identification (since it is interpolatory). However, this approach leads to high order dynamic (the order of the central interpolator coincides with the number of data points used in the identification), necessitating the use of a model reduction step. The resulting identification error leads to the position prediction error. On the other hand, this effect is not present when using the receding horizon filter, since it automatically identifies the lowest order dynamics consistent with the experimental data record.

IV. Structure Recovery from Dynamics

When tracking an unknown number N_o of moving objects, it is of interest to identify (i) the number of objects, (ii) the individual dynamics and, (iii) assign points in the image to each. To illustrate the issues involved, start by considering P features from a single rigid object, tracked over F frames with image coordinates $\{(u_t^p, v_t^p)\}$, $p = 1, \ldots, P$, $t = 1, \ldots, F$. Define the measurement matrix $W_{1:F}$, by:

$$\mathcal{W}_{1:F} = \begin{bmatrix} u_t^p - u_t \\ v_t^p - v_t \end{bmatrix} \in R^{2P \times F} \tag{7}$$

where (u_t, v_t) denote coordinates of the centroid of the features. Under the assumptions of affine projection it can be shown [43] that $\mathcal{W}_{1:F}$ has at most rank 3 and can be decomposed into a rotation $R_{1:F}$ and a "structure" matrix S

$$\mathcal{W}_{1:F} = \begin{bmatrix} R_{1:F}^u \\ R_{1:F}^v \end{bmatrix} S = R_{1:F} S \tag{8}$$

In the case of multiple objects, the number of objects and the corresponding geometry can be obtained by factoring $\mathcal{W}$ into rank 3 submatrices. This basic idea lies at the core of factorization based approaches (see for instance [49], [47]), leading to computationally efficient solutions. However, these approaches cannot disambiguate objects that partially share motion modes, such as the same–wing propellers of the airplane shown in Figure 7(a). It can be easily shown that in this case rank$(\mathcal{W}) = 6$. Hence, as shown in Figure 7 (b)–(c), any motion segmentation approach based solely on finding linearly independent subspaces of the column space of $\mathcal{W}$ will fail, since it cannot distinguish this case from the case of two independently moving propellers. Intuitively, the main difficulty here is that any approach based on properties of $\mathcal{W}$ that are invariant under column permutations, *take into account only geometrical constraints, but not dynamical ones*.

As we show next, robustness can be substantially improved by exploiting the fact that points on the same rigid share more modes of motion than points on different objects. Specifically, begin by associating to the j^{th} object, its centroid $\mathbf{O}^{(j)}$ and an affine basis $b^{(j)}$, centered at $\mathbf{O}^{(j)}$, defined by three no coplanar vectors $\mathbf{V}_i^{(j)}$. Finally, denote by $o^{(j)}(k)$, $v_i^{(j)}(k)$ the coordinates of the image of $\mathbf{O}^{(j)}(k)$ and the projections of $\mathbf{V}_i^{(j)}(k)$ onto the image plane, respectively. Given any point $\mathbf{P}_i^{(j)}$ belonging to the j^{th} object, the coordinates at time k of its image $\mathbf{p}^{(i)}(k)$ are given by:

$$\mathbf{p}_i^{(j)}(k) = \mathbf{o}^{(j)}(k) + \alpha_i^{(j)}\mathbf{v}_1^{(j)}(k) + \beta_i^{(j)}\mathbf{v}_2^{(j)}k + \gamma_i^{(j)}\mathbf{v}_3^{(j)}(k) \tag{9}$$

where $\alpha_i^{(j)}, \beta_i^{(j)}$ and $\gamma_i^{(j)}$ are the *affine invariant* coordinates of $\mathbf{P}_i^{(j)}$ with respect to the basis $b^{(j)}$. Note that, for any two points $\mathbf{P}_r^{(j)}, \mathbf{P}_s^{(j)}$ in the same object, the dynamics of $\mathbf{o}^{(j)}$ are *unobservable* from $\delta_{r,s}(k) \doteq \mathbf{p}_r^{(j)}(k) - \mathbf{p}_s^{(j)}(k)$. Thus, the underlying subsystem is rank deficient when compared to a subsystem describing difference between points on different objects. Roughly speaking, the relative motion of points in a given object, carries no information about the motion of other objects. It follows that points can be clustered in objects according to the complexity of the model required to explain their relative motion. In turn, the order of this model can be estimated by simply computing the rank of the Hankel matrix constructed from the pair-wise differences $\delta_{rs}(k)$, leading to the simple segmentation Algorithm 3, computationally no more expensive than a sequence of SVDs, given below.

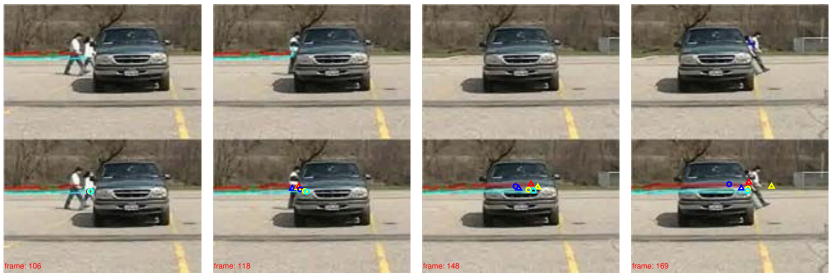


Fig. 6. Tracking two individuals through occlusion using a moving camere. (a-d) frame 106, 118, 148, and 169. 'Triangle' denotes the faster person, 'Circle' denotes the slower person. 'Blue' denotes the track predicted by a Kalman Filter, 'yellow' denotes the track predicted by the combination CF-PF, and 'red' and 'cyan' denote the tracks (one for each target) predicted by the RHRMF. Frame 169 compares the final position estimated by each method against the ground truth.

Algorithm 3: DYNAMICS BASED SEGMENTATION

```
Input.   (i)  $\mathcal{W}$:   the measurements matrix, where
                  $\mathbf{w}_t^i = \begin{bmatrix} u_t^i \\ v_t^i \end{bmatrix}$ is the $i^{th}$ point
                  position in the $t^{th}$ frame.
              $N_p$:   number of features.
              $N_F$:   number of frames.
         (ii) $\sigma_n$:   noise standard deviation.
Output.       $\mathbf{\Gamma}$:   Sorted coupling matrix.

    for all $i \neq j \in \{1, \cdots, N_p\}$ do
        $\mathbf{H} \leftarrow \begin{bmatrix} d_1 & d_2 & \cdots & d_{\frac{N_F}{2}} \\ d_2 & d_3 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ d_{\frac{N_F}{2}} & \cdots & \cdots & d_{N_F} \end{bmatrix}$
            where $d_t = \begin{bmatrix} \mathbf{w}_t^i - \mathbf{w}_t^j \end{bmatrix}$
        Compute $\mathbf{H} = \mathbf{UDV}^T$ using SVD.
        $\mathbf{\Gamma}_{ij} \leftarrow$ number of singular values $\geq \sigma_n$ .
    end for
    reorder $\mathbf{\Gamma}$ using the approach in [5]
```

The effectiveness of this approach is illustrated in Figure 7(d), showing that it correctly identified the presence of four independently moving objects. For comparison, methods relying solely on factorizations of $\mathcal{W}$ [49], [47] fail to correctly segment the objects as seen in Figure 7(b) and (c).

An interesting property of this algorithm is that it allows for a hierarchical motion segmentation of non-rigid objects according to the complexity of the dynamics needed to explain the motions – i.e. features can be grouped together in clusters of increasing size if higher order dynamics are tolerated. This is illustrated in Figure 8 where the palms, fingers and hands are hierarchically segmented using dynamics of increasing complexity, without merging features from the two hands, even though they come close to each other in several frames.

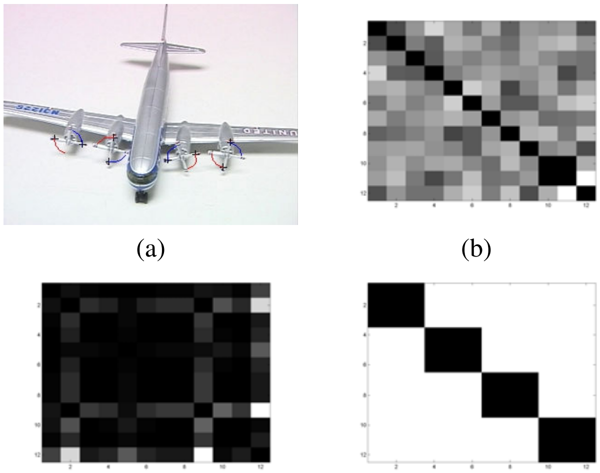

(a) (b) (c) (d)

Fig. 7. (a) Propeller tracks. (b) Costeira-Kanade motion segmentation. (c) Zelnik-Manor-Irani motion segmentation using six eigenvectors. (d) Dynamics based segmentation.

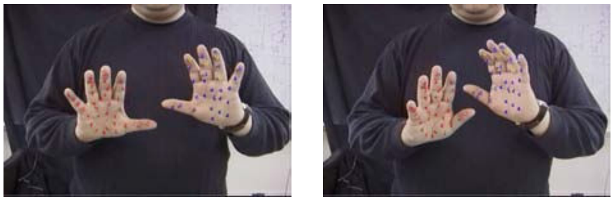


Fig. 8. Hierarchical dynamics-based segmentation of two nonrigid objects.

V. VIDEO INPAINTING AS A RANK MINIMIZATION PROBLEM

Video inpainting, that is the process of seamlessly restoring or altering portions of a video clip, has been the subject of considerable attention in the past few years (see for instance [35] and references therein), but the problem is far from solved. Existing algorithms are limited in the types of sequences that can handle and have relatively high computational complexity. Next, we briefly outline how the use of system theoretic ideas can lead to simple, computationally efficient algorithms that exploit (global) spatio–temporal information. The main idea is to (i) find a set of descriptors that encapsulate the information necessary to reconstruct a frame, (ii) find an optimal estimate of the value of these descriptors for the missing/corrupted frames, and (iii) use the estimated values to reconstruct the frames. In turn, the optimal descriptor estimates can be efficiently obtained postulating that the correct values of the missing descriptors are such that the resulting inpainted sequence is described by the simplest possible (eg. lowest order) dynamical model[3]. Since the order of the underlying model can be estimated from the Hankel matrix of the data, this idea leads again to a rank minimization problem, which in turn can be relaxed to an LMI optimization, solved in Algorithm 2. In this case ζ_i^* denotes either the observed data y_i, if the i frame is present, or the unknown value x_i, if the frame needs to be inpainted. The potential of this approach is illustrated in Fig. 9, where it was used to restore the occluded person. In this particular example, the positions of the 6 feature points indicated in the figure were chosen as descriptors. The video has 36 frames, and occlusion occurs in frames 17 through 19. Using the algorithm outlined above implemented in MATLAB to inpaint the missing descriptors required approximately 20 seconds on a P-III 1.2G PC.

VI. Change Detection using Sequential Sparsification

Change detection is a very general concept that is encountered in many areas of computer vision. From edge detection to video segmentation or image segmentation, a variety of computer vision tasks can be considered as change detection problems with different interpretations of *change*. Hence, a general purpose change detection method with only a few adjustable parameters is very valuable.

Under the assumption that there exists an underlying piecewise affine model for the data (e.g. vectors are clustered in different subspaces), the main objective is to find when the model changes from one mode to another and, at the same time, learn the parameters of the model. Hybrid piecewise affine models [45], [19] and mixture models [3], [42], [38] have been the object of considerable attention in the past few years. Although some of the work (for instance [3]) assumes a fixed number of models, one of the main problems when working with hybrid models is that the number of models is usually unknown. [45] provides a closed form algebraic solution for the noise free case, but the estimation of the number of models usually fails when the data is noisy. More recently, [30] provides an approach exploiting recent results on signal sparsification that is more robust to noise.

Change detection using sequential sparsification [30] seeks to find the minimum number of clusters – i.e. the *simplest* model representing the data – while exploiting the sequential nature of the data. For example, neighboring pixels in an image or consecutive frames in a video sequence are more likely to be within the same segment, and thus imposing continuity of the clusters leads to improved robustness.

The main idea is to robustly search for models that explain the observed data with the lowest possible number of switches (e.g. looking for segmentations that maximize the length of subsequences). This, in turn, is equivalent to searching for descriptions that maximize the *sparsity* of the vector of first order temporal parameter differences, since each non-zero element of this vector corresponds to a switch. Maximizing sparsity is a combinatorial problem and it is generally NP-Hard. However, recent developments show that l_1-norm minimization provides a very good approximation for sparse signal recovery. Moreover, as shown in [10] and [44], this relaxation is indeed exact in the case where the constraints form an underdetermined linear system.

To formalize the problem, consider an affine parametric hybrid model with unknown parameters of the form:

$$\mathcal{H} : f\left(\mathbf{p}_{\sigma(t)}, \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = \mathbf{0} \tag{10}$$

where f is an affine function[4] of the parameter vector $\mathbf{p}_{\sigma(t)}$ which takes values from a finite unknown set according to a piecewise constant function $\sigma(t)$. Here i and j are positive integers that account for the memory of the model (e.g. $j = 0$ corresponds to a causal model, or $i = j = 0$ corresponds to a memoryless model) and we say that there exists a *change* at time t if $\sigma(t) \neq \sigma(t+1)$.

Then, segmentation of a given sequence $\left\{\mathbf{x}(t) \in \mathbb{R}^d\right\}_{t=1}^T$ generated by a hybrid parametric model $\mathcal{H}$ of the form (10) into subsequences is equivalent to finding how many times and when these changes occur. To accomplish this one can consider the sequence of *first order differences* of the parameters $\mathbf{p}(t)$, given by

$$\mathbf{g}(t) = \mathbf{p}(t) - \mathbf{p}(t+1) \tag{11}$$

Clearly, since a non-zero element of this sequence corresponds to a *change*, the sequence should be sparse having only $N-1$ non-zero elements out of T. Furthermore, noise can be taken into account by introducing a noise term $\eta(t)$, satisfying $\|\eta\|_* \leq \epsilon$, where $\|.\|_*$ denotes a norm relevant to the specific problem under consideration and ϵ is an upper bound on the noise level. In this context, change detection can be recast as an optimization problem as follows[5]:

$$\begin{array}{rl} \text{minimize}_{\mathbf{p}(t),\eta(t)} & \|\{\mathbf{g}\}\|_{l_0} \\ \text{subject to} & f\left(\mathbf{p}(t), \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = \eta(t) \quad \forall t \\ & \|\{\eta\}\|_* \leq \epsilon \end{array} \tag{12}$$

Here l_0 is a quasinorm that counts non-zero elements (i.e. minimizing l_0 norm is the same as maximizing sparsity) and can be approximated by the l_1 norm, leading to a linear cost function. When f is an affine function of $\mathbf{p}(t)$, (12) has a convex feasibility set $\mathcal{F}$. Thus, using the l_1 norm leads to a convex, computationally tractable relaxation. Further, Fazel *et al.* proposed an iterative procedure in [13] and [25] to improve the solution obtained by the l_1-norm relaxation where, at each iteration, it solves the following weighted l_1-norm minimization on the convex feasible set $\mathcal{F}$:

[3] It can be shown that this is indeed the case for periodic sequences, but empirical results show that this hypothesis works well also for non–periodic textures.

[4] That is: $f\left(\mathbf{p}_{\sigma(t)}, \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = A(\mathbf{x})\mathbf{p}_{\sigma(t)} + \mathbf{b}(\mathbf{x})$

[5] If $f(\mathbf{0}, \cdot)$ is the zero function, (12) has a trivial solution $\mathbf{p}(t) = 0$ for all t. This problem can be overcome by working with models where $f(\mathbf{0}, \cdot)$ is not the zero function.

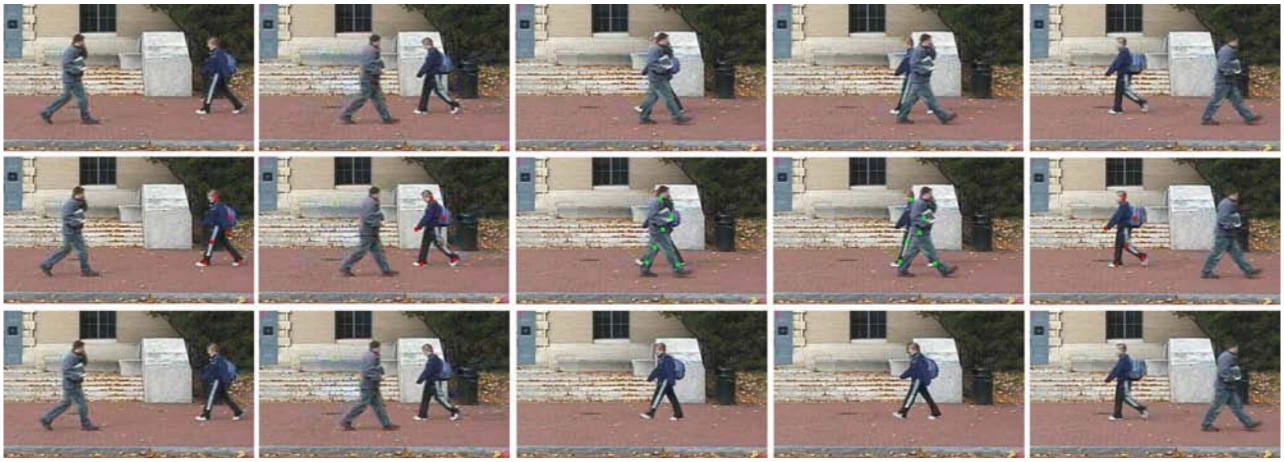


Fig. 9. Top: original sequence. Middle: observed and estimated descriptors. Bottom: inpainted sequence.

$$\begin{array}{lll} \text{minimize}_{z,g,p,\eta} & \sum_{t=1}^{T-1} w_t^{(k)} z_t & \\ \text{subject to} & \|\mathbf{g}(t)\|_\infty \leq z_t & \forall t \\ & f\left(\mathbf{p}(t), \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = \eta(t) & \forall t \\ & \|\{\eta\}\|_* \leq \epsilon & \end{array} \tag{13}$$

where $w_i^{(k)} = (z_i^{(k)} + \delta)^{-1}$ are weights with $z_i^{(k)}$ being the arguments of the optimal solution at the k^{th} iteration and $z^{(0)} = [1, 1, .., 1]^T$; and where δ is a (small) regularization constant that determines what should be considered zero.

The choice of $*$, the norm characterizing the noise, is application dependent. For instance the l_∞-norm performs well in finding anomalies, since in this case the change detection algorithm looks for *local* errors, highlighting outliers. On the other hand, when a bound on the l_1 or l_2-norm of the noise is used, the change detection algorithm is more robust to outliers and it favors the continuity of the segments (i.e. longer subsequences). In addition, when using these norms, the optimization problem automatically adjusts the noise distribution among the segments, better handling the case where the noise level is different in different segments.

A. Video Segmentation

Segmenting and indexing video sequences have drawn a significant attention due to the increasing amounts of data in digital video databases. Systems that are capable of segmenting video and extracting key frames that summarize the video content can substantially simplify browsing these databases over a network and retrieving important content. An analysis of the performances of early shot change detection algorithms is given in [17]. The methods analyzed in [17] can be categorized into two major groups: i) methods based on histogram distances, and ii) methods based on variations of MPEG coefficients. A comprehensive study is given in [48] where a formal framework for evaluation is also developed. Other methods include those where scene segmentation is based on image mosaicking [32], [36] or frames are segmented according to underlying subspace structure [26].

Given a video sequence of frames $\left\{\mathcal{I}(t) \in \mathbb{R}^D\right\}_{t=1}^T$, the video segmentation problem can be solved by applying the sparsification algorithm to the projection of the data into a lower dimensional space PCA (to exploit the fact that the number of pixels D is usually much larger than the dimension of the subspace where the frames are embedded):

$$\mathcal{I}(t) \longmapsto \mathbf{x}(t) \in \mathbb{R}^d.$$

Assuming that each $\mathbf{x}(t)$ within the same segment lies on the same hyperplane not passing through the origin[6] leads to the following hybrid model:

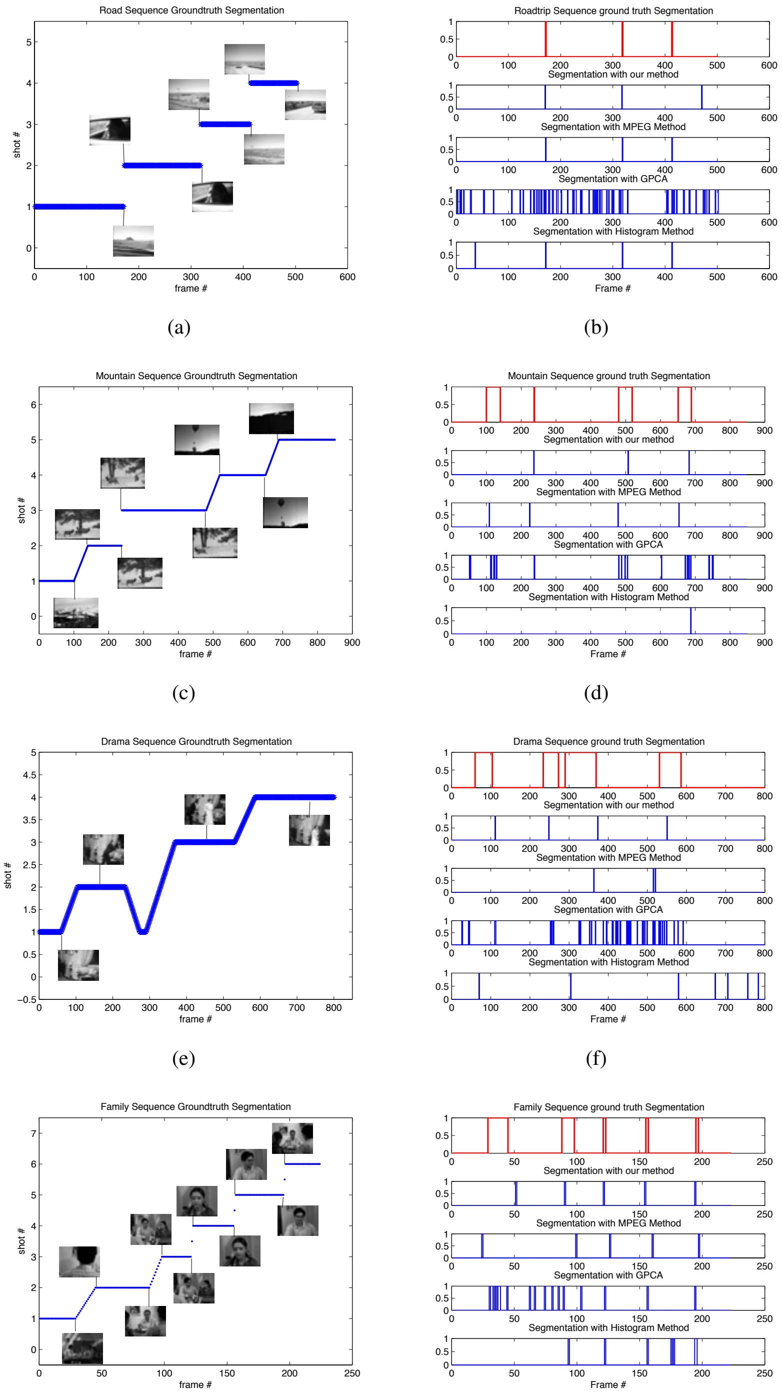
$$\mathcal{H}_1 : f\left(\mathbf{p}_{\sigma(t)}, \mathbf{x}(t)\right) = \mathbf{p}_{\sigma(t)}^T \mathbf{x}(t) - 1 = 0 \tag{14}$$

Thus, in this context algorithm (13) can be directly used to robustly segment the video sequence. It is also worth stressing that as a by-product this method also performs *key frame extraction* by selecting $\mathcal{I}(t)$ corresponding to the minimum $\|\eta(t)\|$ value in a segment (e.g. the frame with the smallest fitting error) as a good representative of the entire segment.

The content of a video sequence usually changes in a variety ways: For instance: the camera can switch between different scenes (e.g. shots); the activity within the scene can change over time; objects or people can enter or exit the scene, etc. There is a hierarchy in the level of segmentation one would require. The noise level ϵ can be used as a tuning knob in this sense.

Figure 10 shows the results of applying this approach to four video sequences (`roadtrip.avi`, `mountain.avi`, `drama.avi` and `family.avi`) available from `http://www.open-video.org`. The original mpeg files were decompressed, converted to grayscale and title frames were removed. Each sequence shows a different characteristic on the transition from one shot to the other. The camera is mostly non-stationary, either shaking or moving. For comparison, results using GPCA, a histogram

[6] Note that this always can be assumed without loss of generality due to the presence of noise in the data.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 10. Video Segmentation Results. Left Column: Ground truth segmentation. Right Column: Changes detected with different methods. Value 0 corresponds to frames within a segment and value 1 corresponds to the frames in transitions.

	Roadtrip	Mountain	Drama	Family
Sparsification	0.9373	0.9629	0.9802	0.9638
MPEG	1	0.9816	0.9133	0.9480
GPCA	0.6965	0.9263	0.7968	0.8220
Histogram	0.9615	0.5690	0.8809	0.9078

TABLE I

RAND INDICES

based method and an MPEG method for segmenting the sequences with optimal parameters (found by trial and error) are also shown. Table I shows the Rand indices [37] corresponding to the clustering results obtained using the different methods, providing a quantitative criteria for comparison. Since the Rand index does not handle dual memberships, the frames corresponding to transitions were neglected while calculating the indices. These results show that indeed the sparcity method does well, with the worst relative performance being against MPEG and B2B in the sequence Roadtrip. This is mostly due to the fact that the parameters in both of these methods were adjusted by a lengthy trial and error process to yield optimal performance in this sequence. Indeed, in the case of MPEG based segmentation, the two parameters governing cut detection were adjusted to give optimal performance in the Roadtrip sequence, while the five gradual transition parameters were optimized for the Mountain sequence.

B. Segmentation of Dynamic Textures

Modeling, recognition, synthesis and segmentation of dynamic textures have drawn a significant attention in recent years [11], [3], [4], [18]). In the case of segmentation tasks, the most commonly used models are mixture models, which are consistent with the hybrid model framework.

In the sequential sparsification framework, the problem of temporal segmentation of dynamic textures reduces to the same mathematical problem as the video segmentation problem, with the difference that now the underlying hybrid model should take the dynamics into account. First, dimensionality reduction is performed via PCA ($\mathcal{I}(t) \longmapsto \mathbf{y}(t) \in \mathbb{R}^d$) and then the reduced-order data is assumed to satisfy a simple causal autoregressive model similar to the one in [4]. Specifically, in this case the hybrid model is given by:

$$\mathcal{H}_2 : f\left(\mathbf{p}_{\sigma(t)}, \{\mathbf{y}(k)\}_{k=t-n}^{t}\right) = \mathbf{p}_{\sigma(t)}^T \begin{bmatrix} \mathbf{y}(t-n) \\ \vdots \\ \mathbf{y}(t) \end{bmatrix} - 1 = 0 \tag{15}$$

where n is the regressor order. This model, which can be considered as a step driven ARX model, was found to be effective experimentally[7]. The power of this approach is illustrated in figure 11 where two very challenging sequences were segmented. The first sequence consists of a patch of dynamic texture (smoke) appended in time to another patch from the same texture but transposed. Thus, the two subsequences have the same photometric properties but differ in the main motion direction. The second sequence was generated using another dynamic texture (river) by sliding a window both in space and time (by going forward in time in the first half and by going backward in the second), thus reversing the dynamics due to the river flow.

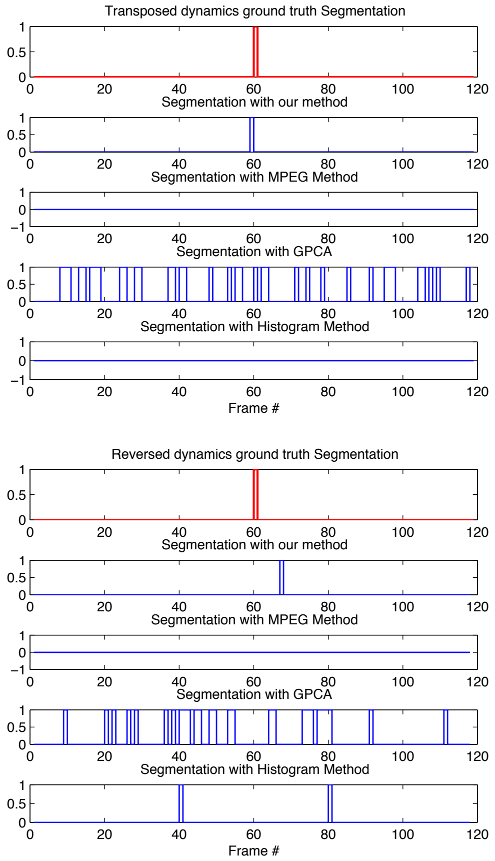


Fig. 11. Results for detecting change in dynamics only. Top: Smoke sequence concatenated with transposed dynamics. Bottom: River sequence concatenated with reversed dynamics.

VII. TEXTURED IMAGE PROCESSING

Texture has been the subject of research in image processing for over three decades, with applications ranging from medical diagnosis to entertainment to human computer interfaces. During the past few years, significant advances have been made in addressing multiple aspects of the problem, ranging from inpainting and synthesis to classification. However, at present, each sub-problem is addressed using a specific set of tailored tools [16]. As we illustrate next, system theoretic tools can lead to a unified framework capable of exploiting the synergism between different aspects of the problem to improve robustness and reduce the computational burden.

[7] The independent term 1 here accounts for an exogenous driving signal. Normalizing the value of this signal to 1, essentially amounts to absorbing its dynamics into the coefficients $\mathbf{p}$ of the model. This allows for detecting both changes in the coefficients of the model and in the statistics of the driving signal.

A. Texture Modeling and Synthesis

Compact models of textured images can be obtained by treating the intensity values $\mathcal{I}(k,l)$ at the (k,l) pixel of the image as the as the output of a *two-dimensional*, discrete linear shift-invariant system driven by white noise, reducing the problem to an identification one: obtaining a model G from image data, possibly corrupted by noise. Note that this requires considering two–dimensional, *non–causal* systems, since the intensity value at a pixel is likely to depend on the values of all pixels in its neighborhood, not just on those preceding it in some ordering of the image pixels. This difficulty can be circumvented by considering a given $n \times m$ image as one period of an infinite 2D signal with period (n,m). Thus, at any given location (i,j) in the image, the intensity values $\mathcal{I}(r,s)$ at other pixels are available also at position $(r-qn, s-qm)$, and the integer q can always be chosen so that $r-qn < i, s-qm < j$. From this observation, it follows that the unknown system G admits a state space representation of form:

$$\begin{aligned} x'(i,j) &= Ax(i,j) + Bu(i,j) \\ \mathcal{I}(i,j) &= Cx(i,j) + Du(i,j) \end{aligned} \tag{16}$$

where

$$x'(i,j) = \begin{bmatrix} x^v(i+1,j) \\ x^h(i,j+1) \end{bmatrix}, x(i,j) = \begin{bmatrix} x^v(i,j) \\ x^h(i,j) \end{bmatrix}$$

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}$$

subject to an additional constraint of the form

$$\begin{matrix} g(i+N,j) = g(i,j) \\ g(i,j+M) = g(i,j) \end{matrix} \quad \text{for some finite } N, M > 0$$

where $g(.,.)$ denotes the impulse response of G. With these assumptions, the problem becomes one of identifying a state–space realization from experimental data, subject to a periodicity constraint, precisely the type of problems solved in [7]. The potential of this approach is illustrated in Fig. 12, where it was used to expand partial images by first identifying the underlying model and then simply computing its impulse response.

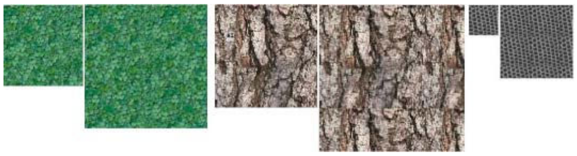


Fig. 12. Using 2-D Models to Expand Images

B. Texture Classification

In this section we show how the models obtained above can be used for texture classification. Proceeding as in [41], we will recast the problem into a robust semi-blind model (in)validation form. To this effect, we will postulate that all images corresponding to realizations of a given texture $\mathcal{T}$ can be obtained as the output of a 2-D operator S to an unknown input signal e with unit spectral density, applied in $(-\infty,0] \times (-\infty,0]$. This leads to the set-up shown in Figure 13, where $T(z_1,z_2)$ represents a nominal model of a particular texture, $h(i,j)$ and $y(i,j)$ denote the intensity value of the ideal and actual images, respectively, and where the (unknown) operator $\Delta(z_1,z_2)$ describes the mismatch between these two images.

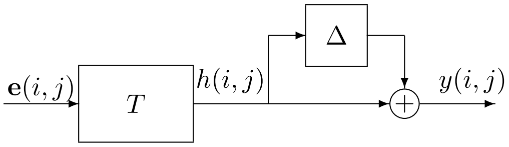
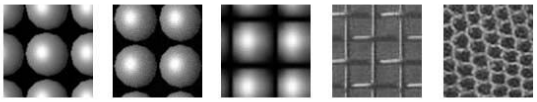


Fig. 13. Texture Recognition Set-up

In this context, given a set of texture families, each represented by a model T_i, an unknown specimen can be classified by (i) performing a sequence of invalidation models to find the lowest uncertainty value $\|\Delta_i\|$ required to explain the specimen in terms of the model T_i, and (ii) assigning the unknown texture to the family corresponding to smallest uncertainty norm. By identifying first a (separable) model of the nominal texture, the corresponding 2-D model invalidation problem can be reduced to two decoupled 1-D semi–blind validation problems that can be solved using the LMI–based technique developed in [41].



$I_f^{1,2}$	$I_g^{1,2}$	$I_f^{1,3}$	$I_g^{1,3}$	$I_f^{1,4}$	$I_g^{1,4}$	$I_f^{1,5}$	$I_g^{1,5}$
0.01	0.01	0.5	0.5	0.7	0.75	0.85	0.83

Fig. 14. Top: Sample Textures. Bottom: Optimal γ

Figure 14 shows the results of applying the technique outlined above to classify several images. Here $I_f^{1,j}$ and $I_g^{1,j}$ denote the results obtained when comparing the decompositions corresponding to the first image against the models obtained from the j^{th} texture. As shown there, this approach correctly indicates that the first three images belong to the same family[8].

VIII. Conclusions

Dynamic vision and imaging is arguably one of the few areas where both further advances and widespread field deployment are being held up not by the lack of a supporting infrastructure, but the lack of *supporting theory*. In this paper paper we illustrated the central role that systems theory can play in developing a comprehensive framework leading to provably robust dynamic vision and imaging systems. In turn, as noted in the introduction, these fields can provide a rich

[8] The higher values of $I_f^{1,3}$ and $I_v^{1,3}$ are due to the use of a lower quality image for the third texture.

environment both to draw inspiration from and to test new developments in systems theory.

Acknowledgements

The authors are indebted to Drs. B. Franzen, T. Ding, H. Lim, C. Mazzaro and W. Ma, and to V. Morariu, R. Lublinerman and N. Ozay for their contributions to the research reported here.

References

[1] A. Alessandri, M. Baglietto, and G. Battistelli. Receding-horizon estimation for discrete-time linear systems. *IEEE Trans. on Automatic Control*, 48(3):473–478, 2003.

[2] O. Camps, H. Li, M. C. Mazzaro, and M. Sznaier. A caratheodory-fejer approach to robust multiframe tracking". In *Proceedings of ICCV 2003*, volume 2, pages 1048–1055. IEEE, 2003.

[3] A. B. Chan and N. Vasconcelos. Mixtures of dynamic textures. In *ICCV05*, volume 1, pages 641–647, 2005.

[4] L. Cooper, J. Liu, and K. Huang. Spatial segmentation of temporal texture using mixture linear models. In *WDV05*, pages 142–150, 2005.

[5] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, September 1998.

[6] T. Ding, Sznaier M., and O.I. Camps. Receding horizon rank minimization based estimation with applications to visual tracking. In *IEEE Conference on Decision and Control*, page "to appear", Dec 2008.

[7] T. Ding, M. Sznaier, and O. Camps. Robust identification of 2-d periodic systems with applications to texture synthesis and classification. In *45^{th} Conf. Dec. Control*, San Diego, CA, 2006.

[8] T. Ding, M. Sznaier, and O. Camps. A rank minimization approach to fast dynamic event detection and track matching in video sequences. In *Proceedings of CDC 2007*, volume 1, pages 4122–4127. IEEE, 2007.

[9] T. Ding, M. Sznaier, and O. Camps. Fast track matching and event detection. In *IEEE Computer Vision and Pattern Recognition*, June 2008.

[10] D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

[11] G. Doretto, A. Chiuso, Y.N. Wu, and S. Soatto. Dynamic textures. *IJCV*, 51(2):91–109, February 2003.

[12] A. Elgammal and C.S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Computer Vision and Pattern Recognition*, pages 681–688, 2004.

[13] M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *American Control Conference*, 2003.

[14] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of American Control Conf. 2004*, volume 4, pages 3273–3278. AACC, 2004.

[15] G. Ferrari-Trecate, D. Mignani, and M. Morari. Moving horizon estimation for hybrid systems. *IEEE Trans. on Automatic Control*, 47(10):1663–1676, 2002.

[16] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2003.

[17] U. Gargi, R. Kasturi, and S. H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):1–13, 2000.

[18] A. Ghoreyshi and R. Vidal. Segmenting dynamic textures with ising descriptors, arx models and level sets. In *WDV06*, pages 127–141, 2006.

[19] W. Hong, J. Wright, K. Huang, and Y. Ma. Multiscale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing*, 15(12):3655–3671, December 2006.

[20] M. Isard and A. Blake. CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[21] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of ACC 1995*, volume 1, pages 1628–1632. IEEE, 1995.

[22] W. H. Kwon, P. S. Kim, and P. Park. A receding horizon kalman fir filter for linear continuous-time systems. *IEEE Trans. on Automatic Control*, 44(11):2115–2120, 1999.

[23] H. Lim, O. I. Camps, and M. Sznaier. A Caratheodory-Fejer approach to appearance modelling. In *IEEE Computer Vision and Pattern Recognition*, volume 1, pages 301–307, 2005.

[24] K. V. Ling and K. W. Lim. Receding horizon recursive state estimation. *IEEE Trans. on Automatic Control*, 44(9):1750–1753, 1999.

[25] M. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152(1):376–394, 2007.

[26] L. Lu and R. Vidal. Combined central and subspace clustering for computer vision applications. In *International Conference on Machine Learning*, pages 593–600, 2006.

[27] W. Ma, H. Lim, M. Sznaier, and O. Camps. Risk adjusted idenitification of wiener systems. In *45^{th}CDC*, San Diego, CA, 2006.

[28] M. Moonen, B. De Moor, L. Vandenberghe, and J. Vandewalle J. On- and off-line identification of linear state-space models. *International Journal of Control*, 49:219–232, 1989.

[29] V. Morariu and O. I. Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *IEEE Computer Vision and Pattern Recognition*, pages 537–544, 2006.

[30] Ozay N, M. Sznaier, and O. Camps. Sequential sparsification for change detection. In *IEEE Computer Vision and Pattern Recognition*, June 2008.

[31] B. North, A. Blake, M. Isard, and J. Rittscher. Learning and classification of complex dynamics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9):1016–1034, September 2000.

[32] M. Osian and L.J. Van Gool. Video shot characterization. *Machine Vision and Applications*, 15(3):172–177, July 2004.

[33] P. V. Overschee. *Subspace identification for linear systems : theory, implementation, applications*. Kluwer Academic Publishers, 1996.

[34] P. A. Parrilo, R. S. Sanchez Pena, and M. Sznaier. A parametric extension of mixed time/frequency domain based robust identification. *IEEE Trans. Autom. Contr.*, 44(2):364–369, 1999.

[35] K. A. Patwardhan, G. Sapiro, and M. Bertalmio. Video inpainting of occluding and occluded objects. In *Proceedings of ICIP 2005*, volume 2, pages 69–72. IEEE, 2005.

[36] N. Petrovic, A. Ivanovic, and N. Jojic. Recursive estimation of generative models of video. In *CVPR06*, pages I: 79–86, 2006.

[37] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[38] O. Rotem, H. Greenspan, and J. Goldberger. Combining region and edge cues for image segmentation in a probabilistic gaussian mixture framework. In *CVPR07*, pages 1–8, 2007.

[39] R. Sánchez Peña and M. Sznaier. *Robust Systems Theory and Applications*. Wiley & Sons, Inc., 1998.

[40] B. Sinopoli, L. Schenato, M. Franceschetti, K. K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. Aut. Control*, 49(2):1453–1464, 2004.

[41] M. Sznaier, M. C. Mazzaro, and O. Camps. Semi-blind model (in)validation with applications to texture classification. In *44^{th}CDC–ECC'05*, pages 6065–6070, Seville, Spain, 2005.

[42] M.E. Tipping and Bishop C.M. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11:443–482, 1999.

[43] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[44] J.A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

[45] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *PAMI*, 27(12):1945–1959, December 2005.

[46] Z. Wanf, F. Yang, D. W.C. Ho, and X. Liu. Robust finite-horizon filtering for stochastic systems with missing measurements. *IEEE Signal Processing Letters*, 12(6):437–440, 2005.

[47] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *The 8th European Conference on Computer Vision (ECCV 2004)*, May 2004.

[48] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(2):168–186, February 2007.

[49] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorization. In *IEEE Computer Vision and Pattern Recognition*, pages 287–293, 2003.