Sequential Sparsification for Change Detection*

Necmiye Ozay, Mario Sznaier and Octavia I. Camps
Dept. of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115

Abstract

*This paper presents a general method for segmenting a vector valued sequence into an unknown number of subsequences where all data points from a subsequence can be represented with the same affine parametric model. The idea is to cluster the data into the minimum number of such subsequences which, as we show, can be cast as a sparse signal recovery problem by exploiting the temporal correlation between consecutive data points. We try to maximize the sparsity (*i.e. *the number of zero elements) of the first order differences of the sequence of parameter vectors. Each non-zero element in the first order difference sequence corresponds to a change. A weighted l_1 norm based convex approximation is adopted to solve the change detection problem. We apply the proposed method to video segmentation and temporal segmentation of dynamic textures.*

1. Introduction

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, we believe that a general purpose change detection method with only a few adjustable parameters will be valuable. This paper takes a step in this direction by exploiting some recently developed results on signal sparsification.

Under the assumption that there exists an underlying piecewise affine model (e.g. vectors are clustered in different subspaces), our 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 [17, 8] and mixture models [1, 15, 14] have been the object of considerable attention in the past few years. Although some of the work (for instance [1]) 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. [17] 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. As we show in this paper, assuming a bound on the noise level, allows for recasting the problem into a robust optimization form where the objective is to find the minimum number of clusters (*i.e*. the simplest model to represent the data). A second point where our method departs from existing clustering techniques is that we make explicit use of 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 result of the present paper shows that the robust segmentation problem can be recast into a change detection form, where the goal is to detect points where the underlying hybrid model switches modes, or, equivalently, to detect changes in the affine parameters describing the model. In principle, detecting these changes can be hard when the measurements are corrupted by noise. However, as we show in the paper, this can be *robustly* accomplished by searching 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 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 [3] and [16], this relaxation is indeed exact in the case where the constraints form an underdetermined linear system. Finally, [5, 9] have very recently presented some algorithmic results that improve upon the l_1-norm relaxation for general sparsity maximization problems subject to convex feasible sets. Exploiting these results leads to efficient, computationally tractable segmentation algorithms.

*This work was supported in part by NSF grants ITR 0312558, ECS 050166, and IIS 0713003 and by AFOSR grant FA 9550-05-1-0437.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

$\mathbf{x}$	a vector in euclidean space $\mathbb{R}^d$
$\|\mathbf{x}\|_p$	p-norm in euclidean space $\mathbb{R}^d$
$\{\mathbf{x}(t)\}_{t=1}^T$,$\{\mathbf{x}\}$	a vector valued sequence of length T where each $\mathbf{x}(t) \in \mathbb{R}^d$
$\|\{\mathbf{x}\}\|_{l_p}$	l_p norm of a vector valued sequence $\|\{\mathbf{x}\}\|_{l_p} \doteq \left(\sum_{i=1}^T \|\mathbf{x}(i)\|_p^p\right)^{1/p}$
$\|\{\mathbf{x}\}\|_{l_0}$	number of non-zero vectors in the sequence (*i.e*. cardinality of the set $\{t \| \mathbf{x}(t) \neq \mathbf{0}, t \in [1,T]\}$)

Table 1. Notation for Sequential Sparsification

The organization of the paper is as follows. In Section 2, we present the problem set-up, main ideas and the algorithm. Potential application areas are stated in Section 3 together with an overview of previous work in these areas. Section 4 illustrates the proposed method with various examples. Finally, Section 5 concludes the paper with some remarks and directions for future research.

2. Segmentation via Sparsification

In this paper we consider the problem of segmenting vector valued sequences $\{\mathbf{x}(t)\}_{t=0}^T$ that are generated by an affine parametric hybrid model with unknown parameters. Specifically, we consider models of the form:

$$\mathcal{H} : f\left(\mathbf{p}_{\sigma(t)}, \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = \mathbf{0} \tag{1}$$

where f is an affine function[1] 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).

We say that there exists a *change* at time t if $\sigma(t) \neq \sigma(t+1)$. Hence segmentation of a given sequence into subsequences is equivalent to finding how many times and when these changes occur. The segmentation problem can be formally stated as follows:

Problem 1 *Given a sequence* $\left\{\mathbf{x}(t) \in \mathbb{R}^d\right\}_{t=1}^T$ *generated by a hybrid parametric model* $\mathcal{H}$ *of the form* (1) *find the minimum number of segments (i.e. subsequences)* $\{\mathcal{S}_i\}_{i=1}^N$ *where on each* $S_i = \{\mathbf{x}(t)\}_{t=T_i}^{T_{i+1}-1}$, $\sigma(t)$ *is constant and* $T_1 = 1$, $T_{N+1} - 1 = T$.

This is a difficult problem, since neither the segmentation nor the parameters of the hybrid model are known. In order to overcome this difficulty, we consider the sequence of *first order differences* of the parameters $\mathbf{p}(t)$, given by

$$\mathbf{g}(t) = \mathbf{p}(t) - \mathbf{p}(t+1) \tag{2}$$

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. Next, in order to account for noise we introduce 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, Problem 1 can be recast as an optimization problem as follows:

$$\begin{array}{ll} \text{minimize}_{\mathbf{p}(t),\eta(t)} & \|\{\mathbf{g}\}\|_{l_0} \\ \text{subject to} & f\left(\mathbf{p}(t), \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = \eta(t) \quad \forall t \\ & \|\{\eta\}\|_* \leq \epsilon \end{array} \tag{P1}$$

[2] 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 $\mathbf{p}(t)$, (P1) 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 [5] and [9] to improve the solution obtained by the l_1-norm relaxation. In the sequel, we adopt this heuristic to solve Problem (P1). This heuristic solves, at each iteration, the following weighted l_1-norm minimization on the convex feasible set $\mathcal{F}$:

$$\begin{array}{ll} \text{minimize}_{z,g,p,\eta} & \sum_{t=1}^{T-1} w_t^{(k)} z_t \\ \text{subject to} & \|\mathbf{g}(t)\|_\infty \leq z_t \quad \forall t \\ & f\left(\mathbf{p}(t), \{\mathbf{x}(k)\}_{k=t-i}^{t+j}\right) = \eta(t) \quad \forall t \\ & \|\{\eta\}\|_* \leq \epsilon \end{array} \tag{P2}$$

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.

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

[2] If $f(\mathbf{0}, \cdot)$ is the zero function, (P1) has a trivial solution $\mathbf{p}(t) = 0$ for all t. To overcome this problem, in this paper we work with models where $f(\mathbf{0}, \cdot)$ is not the zero function.

3. Applications

3.1. 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 [6]. The methods analyzed in [6] 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 [19] where a formal framework for evaluation is also developed. Other methods include those where scene segmentation is based on image mosaicking [11, 12] or frames are segmented according to underlying subspace structure [10]. Formally, the video segmentation problem can be stated as the following instance of Problem 1:

Problem 2 *Given the frames* $\left\{\mathcal{I}(t) \in \mathbb{R}^D\right\}_{t=1}^{T}$*, find* N *segments (i.e. subsequences)* $\{\mathcal{S}_i\}_{i=1}^{N}$ *where* N *is unknown and* $S_i = \{\mathcal{I}(t)\}_{t=T_i}^{T_{i+1}-1}$ *with* $T_1 = 1$*,* $T_{N+1} - 1 = T$*, are generated by an underlying hybrid model.*

Since the number of pixels D is usually much larger than the dimension of the subspace where the frames are embedded, it is reasonable to project the data to a lower dimensional space using PCA:

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

Assuming that each $\mathbf{x}(t)$ within the same segment lies on the same hyperplane not passing through the origin[3] leads to the following hybrid model:

$$\mathcal{H}_1 : f\left(\mathbf{p}_{\sigma(t)}, \mathbf{x}(t)\right) = \mathbf{p}_{\sigma(t)}^T \mathbf{x}(t) - 1 = 0 \qquad (3)$$

Thus, in this context algorithm (P2) can be directly used to robustly segment the video sequence. It is also worth stressing that as a by-product of our method we can also perform *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.

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

3.2. Segmentation of Dynamic Textures

Modeling, recognition, synthesis and segmentation of dynamic textures have drawn a significant attention in recent years [4, 1, 2, 7]). In the case of segmentation tasks, the most commonly used models are mixture models, which are consistent with our hybrid model framework.

In our sequential sparsification framework, the problem of temporal segmentation of dynamic textures reduces to the same mathematical problem as problem 2, 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 [2]. Specifically, the hybrid model we use is:

$$\mathcal{H}_2 : f\left(\mathbf{p}_{\sigma(t)}, \{\mathbf{y}(k)\}_{k=t-n}^{t}\right) = \mathbf{p}_{\sigma(t)}^T \begin{bmatrix} \mathbf{y}(t-n) \\ \vdots \\ \mathbf{y}(t) \end{bmatrix} - 1 = 0 \qquad (4)$$

where n is the regressor order. This model, which can be considered as a step driven ARX model, was found to be effective experimentally[4].

4. Experiments

4.1. Video Segmentation

To evaluate the proposed method for video segmentation, we used four different 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. We applied sequential subspace identification, GPCA, a histogram based method and an MPEG method for segmenting the sequences. For the first two methods, we preprocessed each frame by downsampling it by four and projecting to $\mathbb{R}^3$ using principal component analysis (PCA). For histogram based method, we used bin to bin difference (B2B) with 256 bin histograms and window average thresholding as suggested in [6]. This method has two different parameters. The MPEG method [18] is based on DC-difference images. This method requires seven different parameters, one of which is very sensitive to the length of the *gradual transitions*. In our experiments we adjusted

[4]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.

the parameters of both methods, by trial and error, to get the best possible results. Hence the resulting comparisons against the proposed sequential-sparsification method correspond to best-case scenarios for both MPEG and B2B.

In the roadtrip sequence, the shot changes are in the form of *cuts*. The first three segments, captured in a moving car, have frames switching between the driver and views of country side through the car windows. The last segment, captured from outside the car, shows the car passing by and moving away so that there is an extreme change in the view angle. Figure 1(b) shows the results for this sequence.

The mountain sequence consists of five shots, connected via three *gradual transitions* and one *cut*. The transitions are in the form of approximately forty frames long dissolving effect. Figure 1(c) shows our groundtruth segmentation together with the initial and final frames of each shot. The results obtained using different methods are shown in 1(d).

While the drama sequence consists of a single shot, the semantic meaning of the sequence changes as the actors and actresses enter and exit the scene. Hence, it is still desirable to segment the video so that the whole story can be summarized by using just one frame from each segment. Figure 1(e) shows the groundtruth segmentation[5] together with some key frames. The sequence starts with an empty room, then an actor enters the empty room during the first transition. The first actor leaves the scene between frames 234 and 273. After approximately 20 frames of empty room, a second actor, the actress and the first actor enter the scene. Hence, three people are in the room during segment 3. In the final transition the second actor exits leaving the first actor and the actress back in the room. The segmentation results for this sequence are also show in 1(f).

The family sequence consists of six shots, connected via *gradual transitions* of different lengths. The sequence and its segmentation are shown in Figure 1(h).

Finally, Table 2 shows the Rand indices [13] 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 proposed 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.

[5]Since the segments are not well defined in this case, the groundtruth segmentation is not unique.

	Roadtrip	Mountain	Drama	Family
Our Method	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 2. Rand indices

Figure 2. Sample dynamic texture patches: water, flame, steam.

Sequence Type	Precision	Recall
Two Different Textures	0.8384	0.9167
Three Different Textures	0.7362	0.6061

Table 3. Results on Dynamic Texture Database

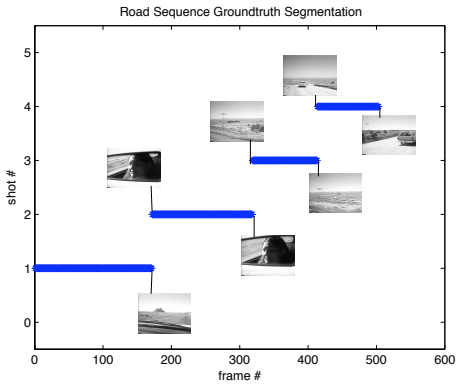
4.2. Temporal Segmentation of Dynamic Textures

For temporal segmentation of dynamic textures, we used the synthetic dynamic texture database (available from `http://www.svcl.ucsd.edu/projects/motiondytex/synthdb/`) to generate a dataset consisting of dynamic textures that change only temporally[6]. We extracted patches of size $35 \times 35 \times 60$ from each segment in the database and concatenated them in time. We applied our algorithm to find the frame number at which the video sequence switches from one texture to another one. Since the number of switches is unknown to our method, there were cases where the method found extra changes or missed an existing change. Table 3 shows the precision and recall rates over a hundred sequences for a fixed noise level. We used fourth order regressors. A change detected within a window of the size of the regressor order from the true frame of change is considered a correct detection.

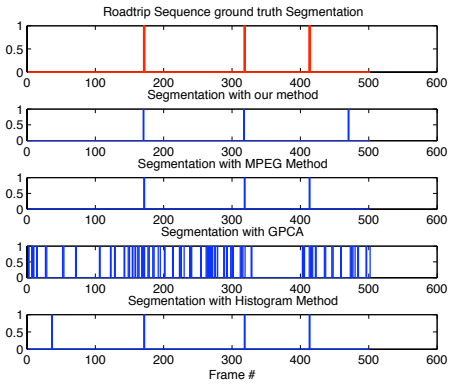
Most of the false positives occured in the sequences that contain flame. This is probably due to the fact that the variance of the stochastic process noise necessary to explain the dynamics of flame is substantially larger than the other textures. Since we used the same noise bound for all dynamic texture experiments, this resulted in extra segments in the sequences that contain flame.

Next, we consider two more challenging sequences. In the first one, we appended in time one patch from smoke to another patch from the same texture but transposed. Therefore, both sequences have the same photometric properties, but differ in the main motion direction: vertical in the first half and horizontal in the second half of the sequence. For

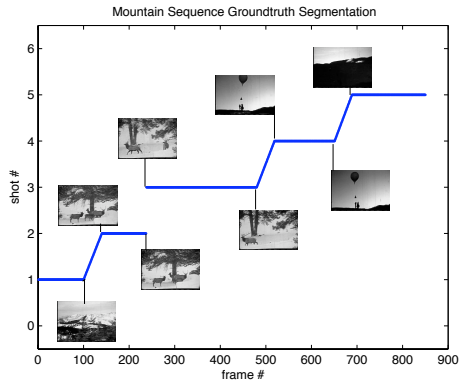
[6]Representative sample patches of these textures are shown in figure 2.


Road Sequence Groundtruth Segmentation
shot #
frame #

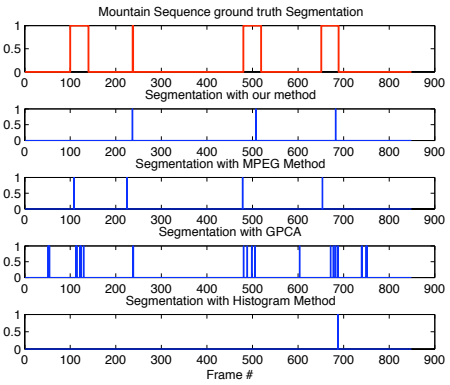
(a)


Roadtrip Sequence ground truth Segmentation
Segmentation with our method
Segmentation with MPEG Method
Segmentation with GPCA
Segmentation with Histogram Method
Frame #

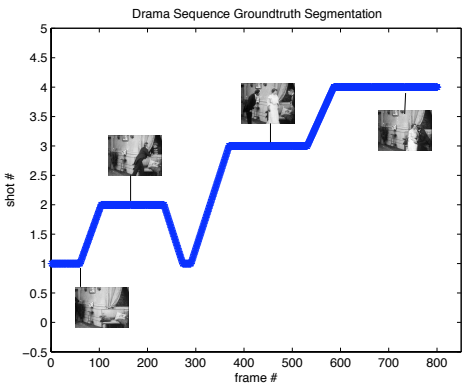
(b)


Mountain Sequence Groundtruth Segmentation
shot #
frame #

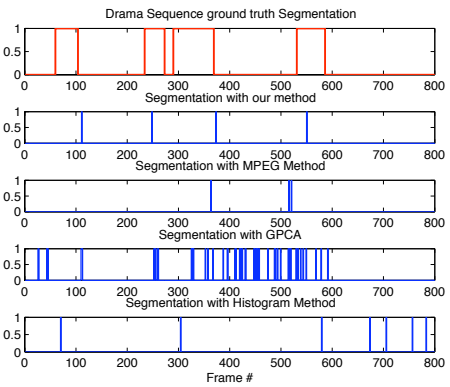
(c)


Mountain Sequence ground truth Segmentation
Segmentation with our method
Segmentation with MPEG Method
Segmentation with GPCA
Segmentation with Histogram Method
Frame #

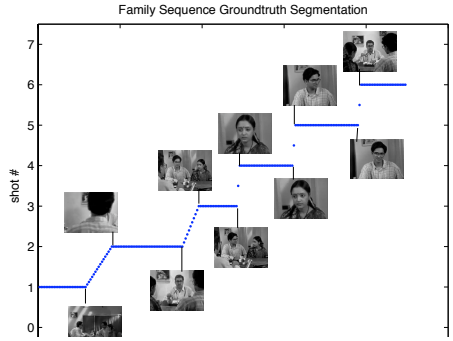
(d)

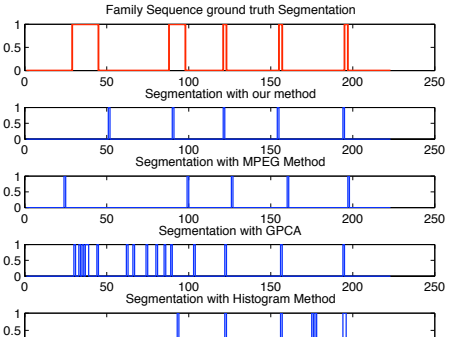

Drama Sequence Groundtruth Segmentation
shot #
frame #

(e)


Drama Sequence ground truth Segmentation
Segmentation with our method
Segmentation with MPEG Method
Segmentation with GPCA
Segmentation with Histogram Method
Frame #

(f)


Family Sequence Groundtruth Segmentation
shot #


Family Sequence ground truth Segmentation
Segmentation with our method
Segmentation with MPEG Method
Segmentation with GPCA
Segmentation with Histogram Method

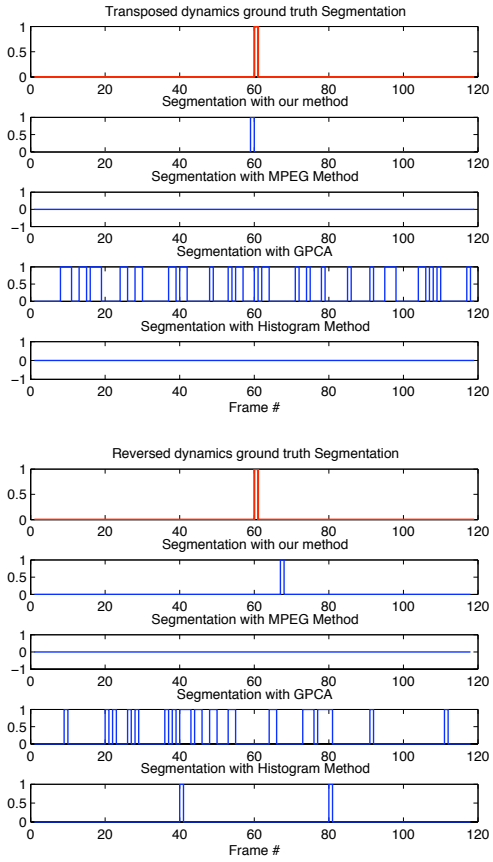


Figure 3. Results for detecting change in dynamics only. Top: Smoke sequence concatenated with transposed dynamics. Bottom: River sequence concatenated with reversed dynamics.

the second example, we generated a sequence of 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). Hence, the dynamics due to the river flow are reversed. For these sequences both the histogram and MPEG methods fail to detect the cut (since the only change is in the dynamics), while the proposed method yields the correct segmentation, as summarized in Figure 3.

5. Conclusions

We proposed a method for segmenting vector valued sequences into a minimum number of subsequences given an underlying hybrid parametric model that is affine in its unknown parameters. As shown both with video and dynamic textures, this method is capable of robust segmentation in the presence of noise, and contrary to many existing methods has few adjustable parameters (essentially just one: the noise level). In addition, the proposed method allows for incorporating any physical insight that may be available about the underlying process. One example is the case of dynamic textures, where it is well-known that they can be modeled as linear stochastic processes. We are currently working on building up reliable models for different segmentation tasks in potential application domains.

References

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

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

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

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

[5] 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 Conf.*, 2003.

[6] U. Gargi, R. Kasturi, and S. H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Trans. Circ. and Syst. for Vid. Tech.*, 10(1):1–13, 2000.

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

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

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

[10] L. Lu and R. Vidal. Combined central and subspace clustering for comp. vision applications. *ICML*, pp. 593–600, 2006.

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

[12] N. Petrovic, A. Ivanovic, and N. Jojic. Recursive estimation of generative models of video. In *CVPR06*, pp. 79–86, 2006.

[13] W. Rand. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.*, 66:846–850, 1971.

[14] 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.

[15] M. Tipping and B. C.M. Mixtures of probabilistic principal component analysers. *Neural Comp.*, 11:443–482, 1999.

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

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

[18] B.-L. Y. and B. Liu. A unified approach to temporal segmentation of motion jpeg and mpeg compressed video. In *Int. Conf. on Multimedia Computing and Sys.*, pp. 81–88, 1995.

[19] J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang. A formal study of shot boundary detection. *IEEE Trans. Circ. and Sys. for Vid. Tech.*, 17(2):168–186, 2007.