Spatio-Temporal Segmentation using Dominant Sets

Andrea Torsello, Massimiliano Pavan, and Marcello Pelillo

Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino 155, 30172 Venezia Mestre, Italy
{torsello,pavan,pelillo}@dsi.unive.it

Abstract. Pairwise data clustering techniques are gaining increasing popularity over traditional, feature-based central grouping techniques. These approaches have proven very powerful when applied to image-segmentation problems. However, they are computationally too demanding to be applied to video-segmentaton problems due to their scaling behavior with the quantity of data. On a dataset containing N examples, the number of potential comparisons scales with $O(N^2)$, thereby rendering the approaches unfeasible for problems involving very large data sets. It is therefore of primary importance to develop strategies to reduce the number of comparisons required by subsampling the data and extending the grouping to out-of-sample points after the clustering process has taken place. In this paper we present an approach to out-of-sample clustering based on the dominant set framework [10] and apply it to video segmentation. The method is compared against two recent pairwise clustering algorithms which provide out-of-sample extensions: the Nyström method [3], and the minimal-shift embedding approach [14]. Our results show that our approach performs comparably against the competition in terms of quality of the segmentation, being, however, much faster.

1 Introduction

Proximity-based, or pairwise, data clustering techniques are gaining increasing popularity over traditional central grouping techniques, which are centered around the notion of "feature" (see, e.g., [5, 15, 16, 14]). In many application domains, in fact, the objects to be clustered are not naturally representable in terms of a vector of features. On the other hand, quite often it is possible to obtain a measure of the similarity/dissimilarity between objects. Although such a representation lacks geometric notions such as scatter and centroid, it is attractive as no feature selection is required and it keeps the algorithm generic and independent from the actual data representation and metric involved. Further, it allows one to use non-metric similarities and it is applicable to problems that do not have a natural embedding to a uniform feature space, such as the grouping of structural or graph-based representations.

These appraoches have proven very powerful when applied to image segmentation problems [16, 7, 5, 2]. However the application of these method to video segmentation is, in general, unfeasible, due to the scaling behavior with the number of data items. On a dataset containing N examples, the number of potential comparisons scales with $O(N^2)$, thereby rendering the approach too demanding, both in terms of computation time and of space, to be used to segment video feeds. A way of overcoming this drawback is to drastically reduce the number of objects to be clustered and then extend the partition to the full data-set. Unfortunately, there is no straightforward way of extending the clustering results to new data within the pairwise grouping paradigm short of recomputing the complete cluster structure.

In an attempt to address this shortcoming of the pairwise approach, Fowlkes et al. [3] have recently proposed to use the Nyström approximation to extend normalized cut to out-of-sample data, while Bengio et al. [1] use the Nyström extension to extend other spectral clustering appraoches. Roth et al. [14] propose to perform pairwise clustering by embedding the distance data in an Euclidean space, and show how this embedding can be extended to new points.

Recently, a new framework for pairwise data clustering based on the graph-theoretic concept of a *dominant set* has emerged [10]. An intriguing connection between dominant sets and the solutions of a (continuous) quadratic optimization problem makes them related in a non-trivial way to spectral-based cluster notions, and allows one to use straightforward dynamics from evolutionary game theory to determine them [17]. A nice feature of this framework is that it naturally provides a principled measure of a cluster's cohesiveness as well as a measure of a vertex participation to its assigned group. The approach has proven to be a powerful one when applied to problems such as intensity, color, and texture segmentation, and is competitive with spectral approaches such as normalized cut [10, 11].

Motivated by the previous arguments, in this paper we address the problem of applying the dominant set approach to spatio-temporal segmentation of video sequences. In order to do this we propose an efficient approach to assigning out-of-sample data to one of a set of previously determined dominant sets. This allows us to substantially reduce the computational burden associated to the processing of the huge amount of data involved in video segmentation. We compare the segmentation obtained with our results to those obtained using the Nyström extension [3] and the minimal-shift embedding method [14], both on video sequences and synthetic data.

2 The dominant set framework

In the pairwise clustering framework the data to be clustered are represented (possibly implicitly) as an undirected edge-weighted graph with no self-loops $G = (V, E, w)$, where $V = \{1, \dots, n\}$ is the vertex set, $E \subseteq V \times V$ is the edge set, and $w : E \rightarrow \mathbb{R}_+^*$ is the (positive) weight function. Vertices in G correspond to data points, edges represent neighborhood relationships, and edge-weights reflect similarity/dissimilarity between pairs of linked vertices. As customary, we represent the graph G with the corresponding weighted adjacency (or similarity/dissimilarity) matrix, which is the $n \times n$ nonnegative, symmetric matrix $A = (a_{ij})$ defined as:

$$a_{ij} = \begin{cases} w(i,j) \ , & \text{if } (i,j) \in E \\ 0 \ , & \text{otherwise .} \end{cases}$$

The dominant set framework has been presented in [10]. Let $S \subseteq V$ be a non-empty subset of vertices and $i \in V$. The *(average) weighted degree* of i w.r.t. S is defined as:

$$\mathrm{awdeg}_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij} \tag{1}$$

where $|S|$ denotes the cardinality of S. Moreover, if $j \notin S$ we define $\phi_S(i,j) = a_{ij} - \mathrm{awdeg}_S(i)$ which is a measure of the similarity between nodes j and i, with respect to the average similarity between node i and its neighbors in S.

Let $S \subseteq V$ be a non-empty subset of vertices and $i \in S$. The *weight* of i w.r.t. S is

$$\mathrm{w}_S(i) = \begin{cases} 1, & \text{if } |S| = 1 \\ \displaystyle\sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(j,i) \, \mathrm{w}_{S \setminus \{i\}}(j) \ , & \text{otherwise} \end{cases} \tag{2}$$

while the *total weight* of S is defined as:

$$\mathrm{W}(S) = \sum_{i \in S} \mathrm{w}_S(i) \ . \tag{3}$$

Intuitively, $\mathrm{w}_S(i)$ gives us a measure of the overall similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall similarity among the vertices in $S \setminus \{i\}$, with positive values indicating high internal coherency.

A non-empty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any non-empty $T \subseteq S$, is said to be *dominant* if:

1. $\mathrm{w}_S(i) > 0$, for all $i \in S$
2. $\mathrm{w}_{S \cup \{i\}}(i) < 0$, for all $i \notin S$.

The two previous conditions correspond to the two main properties of a cluster: the first regards internal homogeneity, whereas the second regards external inhomogeneity. The above definition represents our formalization of the concept of a cluster in an edge-weighted graph.

Now, consider the following quadratic program, which is a generalization of the so-called Motzkin-Straus program [8]:

$$\begin{array}{ll} \text{maximize} & f(\mathbf{x}) = \mathbf{x}' A \mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta_n \end{array} \tag{4}$$

where

$$\Delta_n = \{\mathbf{x} \in \mathrm{I\!R}^n \ : \ x_i \geq 0 \text{ for all } i \in V \text{ and } \mathbf{1}'\mathbf{x} = 1\}$$

is the standard simplex of $\mathbb{R}^n$, and $\mathbf{1}$ is a vector of appropriate length consisting of unit entries. The *support* of a vector $\mathbf{x} \in \Delta_n$ is defined as the set of indices corresponding to its positive components, that is $\sigma(\mathbf{x}) = \{i \in V \ : \ x_i > 0\}$. In [10], an intriguing connection between dominant sets and local solutions of program (4) is established. Specifically, it is proven that if S is a dominant subset of vertices, then its (weighted) characteristic vector $\mathbf{x}^S$, which is the vector of Δ_n defined as

$$x_i^S = \begin{cases} \frac{\mathrm{w}_S(i)}{\mathrm{W}(S)}, & \text{if } i \in S \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

is a strict local solution of program (4). Conversely, if $\mathbf{x}$ is a strict local solution of program (4) then its support $S = \sigma(\mathbf{x})$ is a dominant set, provided that $\mathrm{w}_{S\cup\{i\}}(i) \neq 0$ for all $i \notin S$.

By virtue of this result, a dominant set can be found by localizing a local solution of program (4) with an appropriate continuous optimization technique, such as replicator dynamics from evolutionary game theory [17], and then picking up its support. In order to get a partition of the input data into coherent groups, a simple approach is to iteratively find a dominant set and then remove it from the graph, until all vertices have been grouped.

Note that the components of the weighted characteristic vectors give us a natural measure of the participation or "centrality" of the corresponding vertices in the cluster, whereas the value of the objective function measures the homogeneity or cohesiveness of the class.

3 Out-of-sample extension of a dominant set classification

Suppose we are given a set V of n unlabeled items and let $G = (V, E, w)$ denote the corresponding similarity graph. After determining the dominant sets (i.e., the clusters) for these original data, we are next supplied with a set of m new data items and are asked to assign each of them to one of the previously extracted clusters. A recent approach to the out-of-sample extension of the dominant set framework was presented in [12]. The approach tested each new point against each set to see whether it increased its "cohesiveness". In particular, if $w_{S\cup\{i\}}(i) > 0$, then the new item i was assigned to cluster S. There are, however, a number of problems with this approach. First, it does not provide a partition of the data, since each point can be assigned to more than one cluster or to none at all. More importantly, the approach needs the distances between all the samples and the new item to be avaiable to perform the extension, and has, hence, $O(nm)$ time-complexity, where n is the number of samples and m is the number of out-of-sample items. A similar approach can be found in [4]

In a central-clustering framework a straightforward way to do it is to assign each new vector x to the cluster with the closest centroid, i.e.,

$$C(x) = \operatorname*{argmin}_{\nu}(\|y_\nu - x\|)\,,$$

where y_ν is the centroid of cluster ν. In a pairwise clustering framework, however, the cluster-centroid is not explicit. However, with sufficient samples, we can assume that at least one element of the cluster is "close" to the centroid. Previous experience show that the weight x_i^S of the characteristic vector is a measure of the centrality of item i with respect to the dominat set S. Hence, we take the sample with maximum weight to be the prototype $P(S)$ of S, that is:

$$P(S) = \underset{i}{\operatorname{argmax}}(x_i^S)\,.$$

A similar definition of prototype was recently proposed in [4]. There, however, the prototype was not used for clustering purposes and had no implicit relation to the centroid.

With the prototype at hand, we perform the cluster extension by assigning each new item j to the cluster S_ν with the closest prototype, i.e.,

$$C(j) = \underset{\nu}{\operatorname{argmax}}(\mathrm{Sim}\left(P(S_\nu), j)\right)\,,$$

where $\mathrm{Sim}(i,j)$ is the similarity between items i and j.

The proposed extension is very efficient, since, for each new point to be clustered, it only requires the computation of one distance per cluster. Hence, having a $O(km)$ time complexity, where k is the number of clusters and m the number of out-of-sample items. Furthermore, the extension can be done *on-line* since each new point is assigned to a cluster in isolation, without any information about the similarity structure of the other out-of-sample points.

4 Experimental results

To assess the ability of the proposed approach to perform meaningful segmentation on large data-sets, we apply the algorithm to spatio-temporal segmetnation of video sequences. We compare the performance, both in terms of quality and computation time, to two recent out-of-sample pairwise clustering approaches described in [3] and [14].

4.1 The Nyström method

In a recent paper, Fowlkes et al. propose to use the Nyström Method to extend the Normalized Cut framework to out-of-sample data [3]. The Nyström method is a technique for finding numerical approximations to eigenfunctions problems of the form

$$\int_a^b W(x,y)\phi(y)\,dy = \lambda\hat{\phi}(x)\,.$$

It is based on the idea that this equation can be approximated using a simple quadrature rule on a set of sample points $\zeta_1, \ldots, \zeta_n$ in the interval $[a,b]$:

$$\frac{b-a}{n}\sum_{j=1}^{n} W(x,\zeta_j)\hat{\phi}(\zeta_j) = \lambda\hat{\phi}(x)\,, \tag{6}$$

where $\hat{\phi}(x)$ is the approximation of $\phi(x)$. Let $A_{i,j} = W(\zeta_i, \zeta_j)$ be the matrix obtained by sampling the weight function W at points $\zeta_1, \dots, \zeta_n$, the set $\varPhi = \{\phi_1, \dots, \phi_n\}$ of the eigenvector of A are solutions of the system:

$$\frac{b-a}{n} \sum_{j=1}^{n} W(\zeta_i, \zeta_j)\hat{\phi}(\zeta_j) = \lambda\hat{\phi}(\zeta_i) \quad \forall i \in \{1, \dots, n\}\,.$$

Substituting back into (6) yields the *Nyström extension*

$$\hat{\phi}_i(x) = \frac{1}{n\lambda_i} \sum_{j=1}^{n} W(x, \zeta_j)\phi_i(\zeta_j)\,.$$

Let the complete weight matrix W be

$$W = \begin{bmatrix} A & B \\ B' & C \end{bmatrix}\,,$$

where matrix A holds the similarities between the samples points, B the similarities between sample and out-of-sample points, and C the similarities between out-of-sample points. The Nystöm approach implicitly approximates C with $B'A^{-1}B$, leading to the extended weight matrix

$$\hat{W} = \begin{bmatrix} A & B \\ B' & B'A^{-1}B \end{bmatrix}\,.$$

To apply the Nyström approximation to Normalized Cut, it is necessary to compute the row sums $\hat{d}$of the extended weight matrix $\hat{W}$

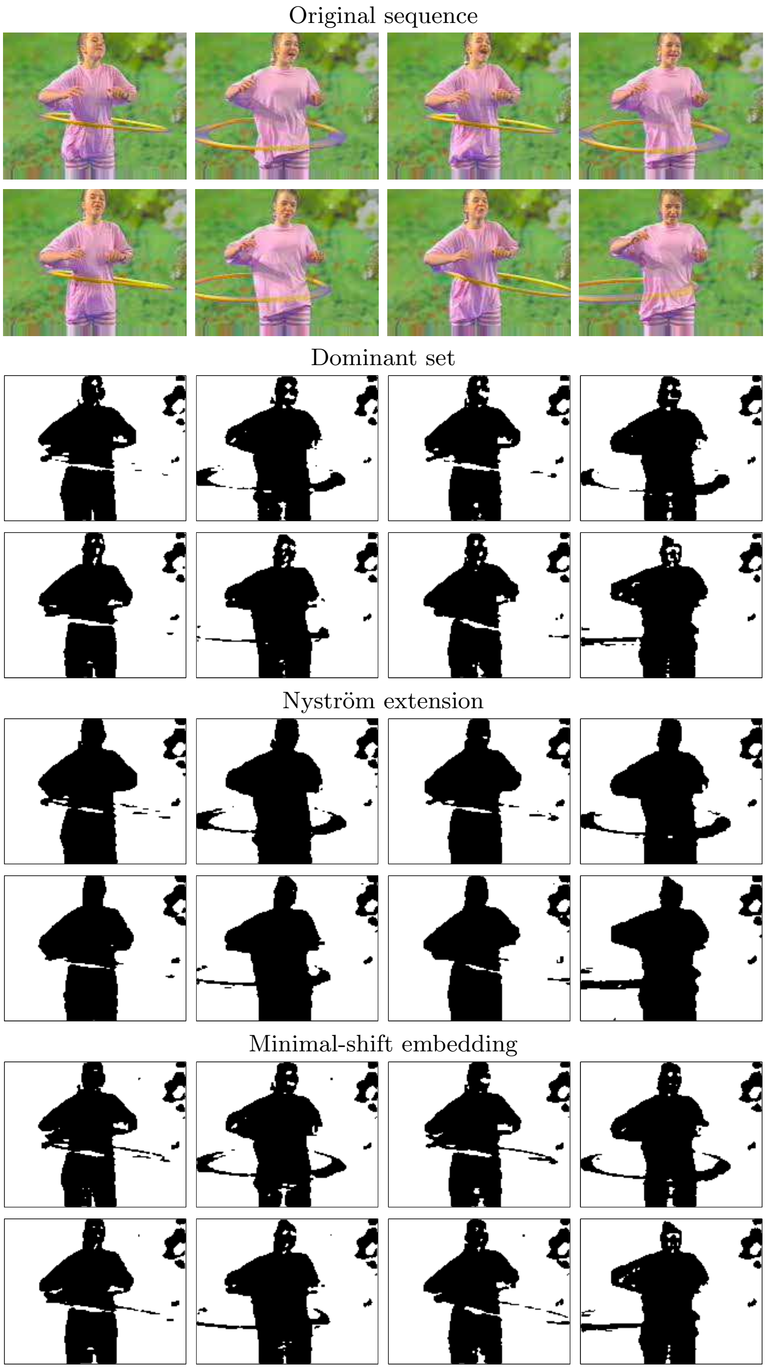
$$\hat{d} = \hat{W}\mathbf{1}_{n+m} = \begin{bmatrix} A\mathbf{1}_n + B\mathbf{1}_m \\ B'\mathbf{1}_n + B'A^{-1}B\mathbf{1}_m \end{bmatrix}\,,$$

where n is the number of samples, m is the number of out-of-sample points. This requires the (implicit) computation of $B\mathbf{1}_m$. Hence the approach cannot easily be expressed as an *on-line* extension and, at least in the implementation provided in [3] and used in our experiments, requires the full matrix B to be in memory. This severely limits the dimension of the video sequences that can be segmented with this approach.

4.2 Minimal-shift embedding

In a recent paper, Roth et al. present an embedding of possibly non-Euclidean distance data that preserve the clustering properties of the k-means functional [14]. The approach derives from the observation that the k-means functional

$$H^{km} = \sum_{\nu=1}^{k} \sum i = 1^n M_{i\nu} \|x_i - y_\nu\|^2\,, \tag{7}$$



Original sequence

Dominant set

Nyström extension

Minimal-shift embedding

Fig. 1. The hula sequence. Top to Bottom: dominant set, Nyström extension, and minimal-shift embedding.

where y_ν is the centroid of cluster ν and $M_{i\nu} = 1$ if item i is assigned to cluster ν, $M_{i\nu} = 0$ otherwise, can be expressed in terms of pairwise distances between the points to be clustered as: In fact, by setting the prototype vectors we can write (7) as

$$H^{km} = \frac{1}{2} \sum_{\nu=1}^{k} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} M_{i\nu} M_{j\nu} \|x_i - x_j\|^2}{\sum_{l=1}^{n} M_{l\nu}} . \tag{8}$$

Furthermore, the authors show that this functional is invariant under a constant shift of the distance matrix D, i.e., $H^{km}(D) = H^{km}(D + d_0(\mathbf{11'} - I))$ where d_0 is an arbitrary constant, $\mathbf{1}$ is the vector of all ones and I is the identity matrix. Let us define the matrix

$$S^c_{ij} = \frac{1}{2} \left[D_{ij} - \frac{1}{n} \sum_{k=1}^{n} D_{ik} - \frac{1}{n} \sum_{k=1}^{n} D_{kj} + \frac{1}{n^2} \sum_{k,l=1}^{n} D_{kl} \right] .$$

Clearly, $D_{ij} = S^c_{ii} + S^c_{jj} - 2S^c_{ij}$. It is a well known result that D derives from a Euclidean distance, or, equivalently, can be embedded in an Euclidean space, if and only if S^c is positive semidefinite. However, the shifted matrix $\tilde{S} = S - \lambda_n T$, where λ_n is the least eigenvector of S, is clearly positive semidefinite, and the distance matrix $\tilde{D}_{ij} = \tilde{S}_{ii} + \tilde{S}_{jj} - 2\tilde{S}_{ij}$ can be obtained from D with a constant shift $\tilde{D} = D - 2\lambda_n(\mathbf{11'} - I)$. Hence, $\tilde{D}$ is embeddable in an Euclidean space and has the same k-means functional as the original matrix D. To perform the clustering, Roth et al. propose to compute the shifted matrix $\tilde{D}$, embed it into an Euclidean space and run the k-means clustering algorithm on the embedding space. To reduce the noise dimensional reduction approaches such as principal component analysis can be applied to the embedding space. This is equivalent to performing multi-dimensional scaling on $\tilde{D}$.

One interesting property of this approach, mentioned in [14], is that it is possible to extend the embedding to new data. First, note that we can write the embedding in the form $X_p = \tilde{S} V_p \Lambda_p^{-1/2} v$, where V_p is the matrix containing the first p eigenvector of $\tilde{S}$ and Λ_p the corresponding eigenvalues matrix. Given the matrix D^n of distances between the new and the old data, we define

$$S^n_{ij} = \frac{1}{2} \left[D^n_{ij} - \frac{1}{n} \sum_{k=1}^{n} D^n_{ik} - \frac{1}{n} \sum_{k=1}^{n} \tilde{D}_{kj} + \frac{1}{n^2} \sum_{k,l=1}^{n} \tilde{D}_{kl} \right]$$

and project the new points as $X^n_p = S^n V_p \Lambda_p$. The new point is then assigned to the cluster with the closest centroid. While this approach allows each new point to be clustered *on-line*, that is without any information about the other out-of-sample points, it requires the computation of the full set of distances between the new point and the original samples to perform the embedding, and hence, the extension is computationally more demanding than using our approach, being of order $O(nm)$, where n is the number of samples and m the number of out-of-sample items.

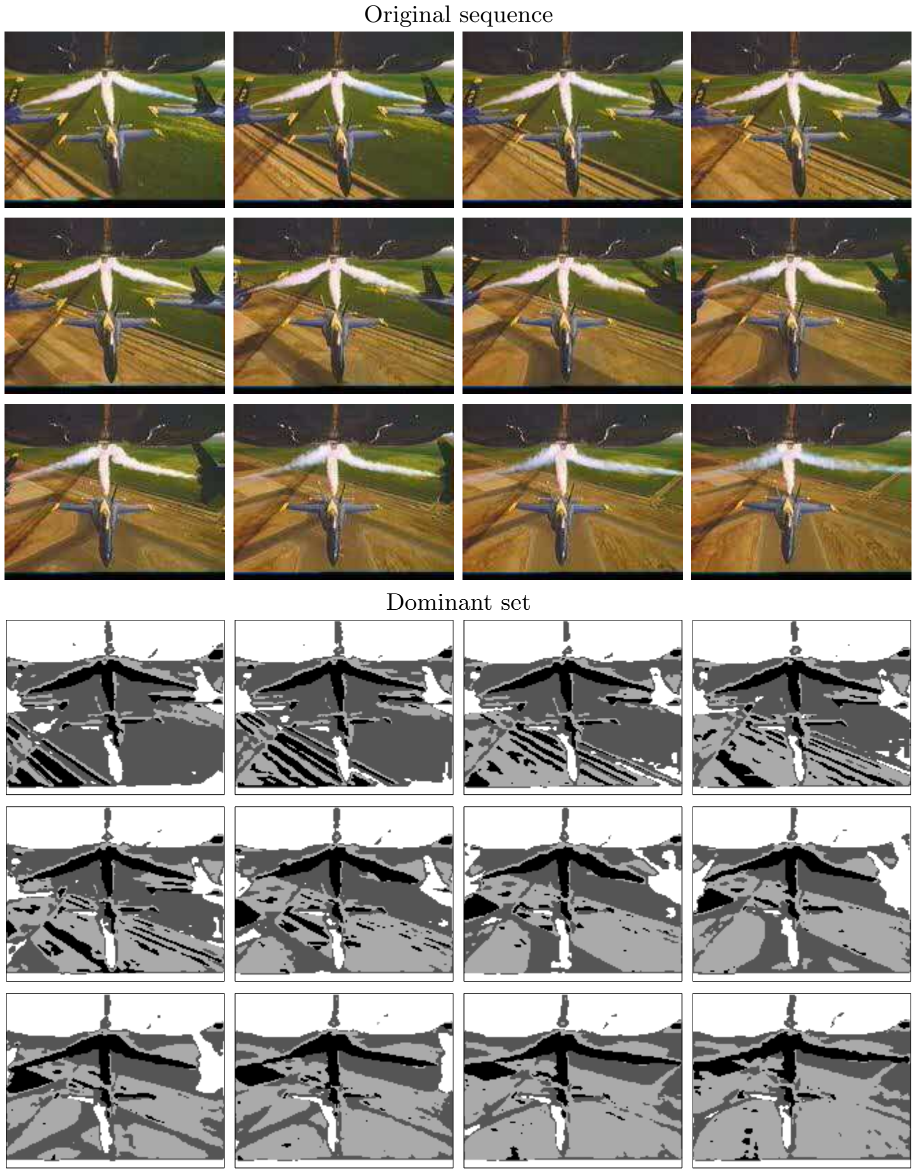
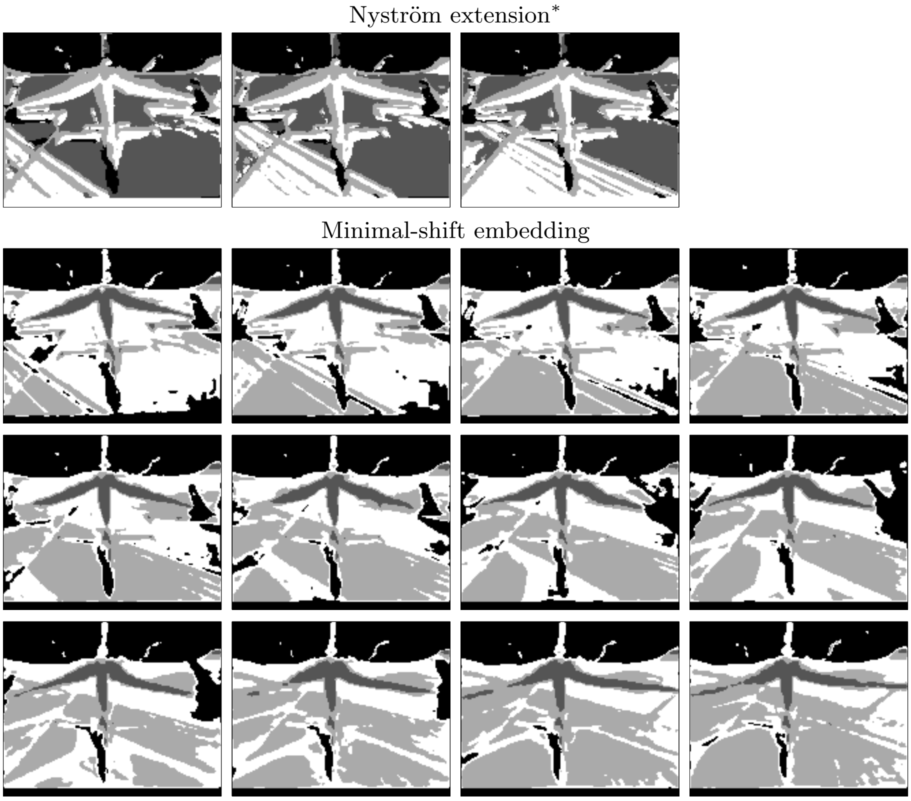


Fig. 2. The flight sequence. Original sequence and the segmentation obtained with the dominant set framework.

4.3 Similarity measures

To measure the similarity/distance between two points in a video sequence we make use of both color and texture information. To measure the difference in texture we convolve each frame with a bank of linear spatial filters. The filters



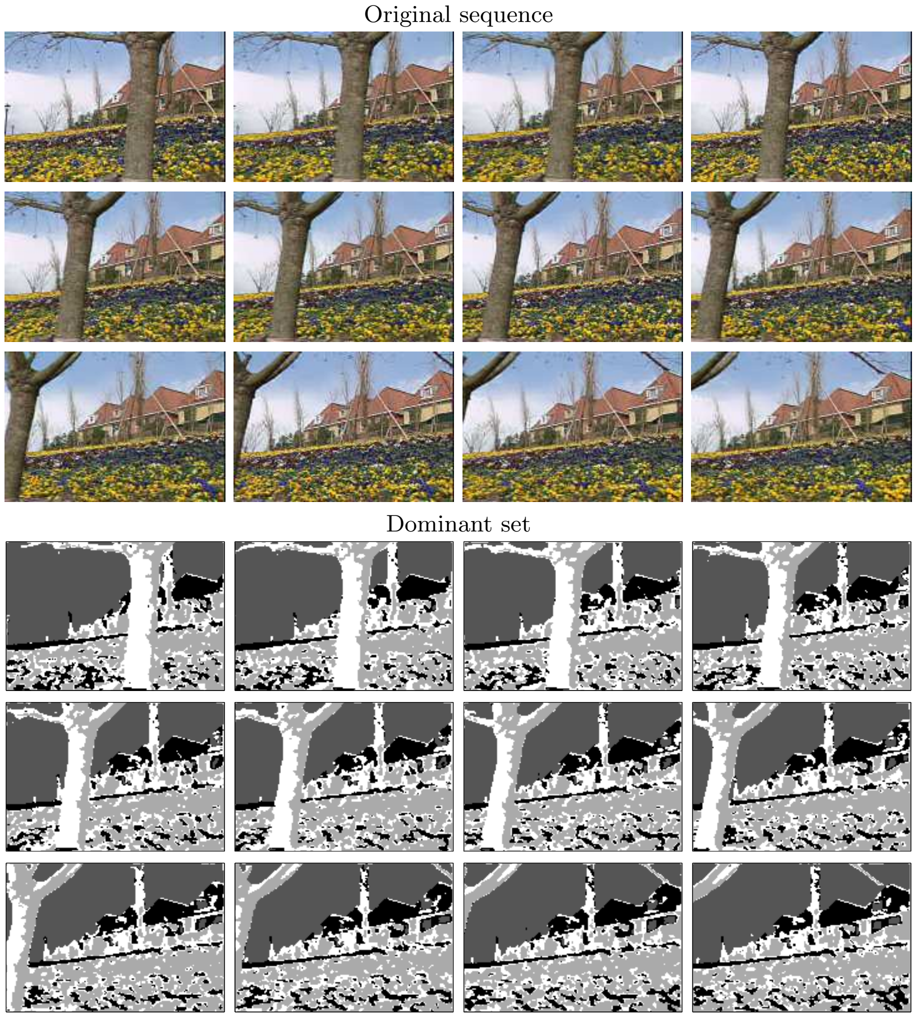
Nyström extension*

Minimal-shift embedding

* Nyström extension could not handle more than 3 frames.

Fig. 3. The flight sequence. Segmentaion obtained with the Nyström extension, and the minimal-shift embedding.

consist of pairs odd- and even-symmetric oriented filters in various scales and orientations, plus a set of center-symmetric filters. The odd-symmetric filters are re-orientation and re-scaling of the base $f_o(x, y) = c_o G'_{\sigma 1}(y) G_{\sigma 2}(x)$, where $G_\sigma(x)$ represents a Gaussian with standard deviation σ and c_o is a constant that forces f_o to unitary L_1 norm. The even-symmetric filters are re-orientation and re-scaling of the base $f_e(x, y) = c_e G'_{\sigma 1}(y) G_{\sigma 2}(x)$, while the center symmetric filters have basis $f_c(x, y) = c_c(G_{\sigma 1}(y) G_{\sigma 1}(x) - G_{\sigma 2}(y) G_{\sigma 2}(x))$. Here the constants c_e and c_c guarantee that all the basis have unitary L_1 norm. Similarly to what was presented in [7], in our experiments we have 6 different orientations uniformly separated by 30^o, and 4 different scales, for a total of 40 filters, hence the filter-responses are vectors in a 40-dimensional space. The difference in texture is measured as the Euclidean distance between the filter-responses, while the difference between two colors is defined as the Euclidean distance of the RGB representations of the two colors. For each two points p and q in the video sequence, we extract their 5x5 spatial neighborhoods and, for each point of in-



Original sequence

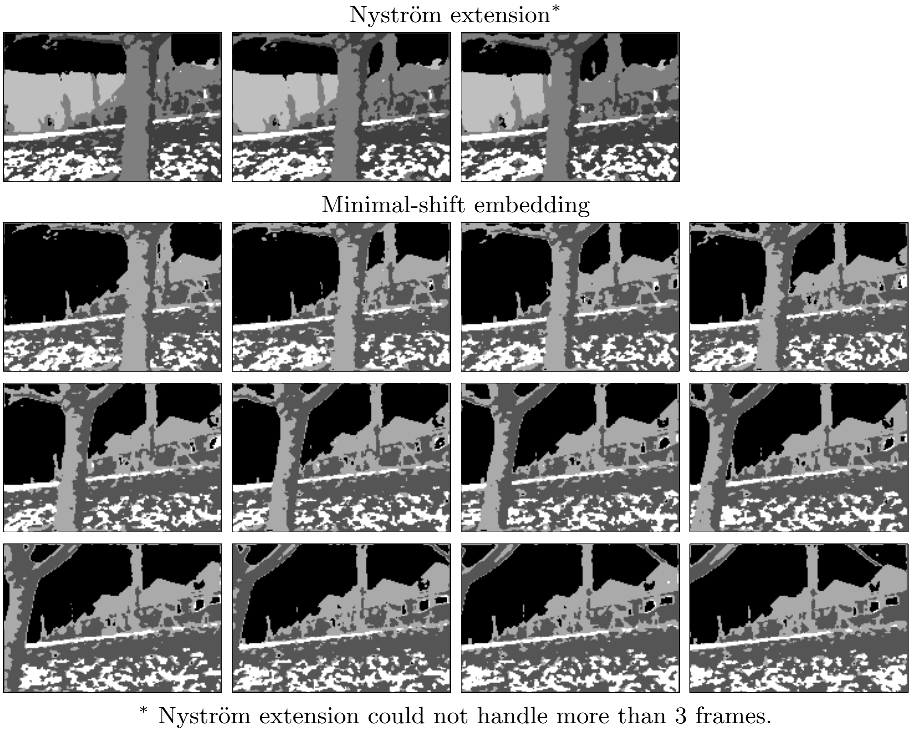
Dominant set

Fig. 4. The flower garden sequence. Original sequence and the segmentation obtained with the dominant set framework.

dex (i, j) in the neighborhoods, we compute the color vectors c_{ij}^p and c_{ij}^q around points p and q respectively. Similarly, we compute the filter-response vectors f_{ij}^p and f_{ij}^q. We define the color-distance function d_c and the filter-distance function d_f as the weighted combination of distances

$$d_c(p,q) = \sum_{i,j=1}^{5} w_{ij} \| c_{ij}^p - c_{ij}^q \| \quad d_f(p,q) \sum_{i,j=1}^{5} w_{ij} \| f_{ij}^p - f_{ij}^q \| \,,$$

where, in our experiments, the weight w_i is a Gaussian centered at the center of the neighborhood and with unitary standard deviation.


Nyström extension*

Minimal-shift embedding

* Nyström extension could not handle more than 3 frames.

Fig. 5. The flower garden sequence. Segmentaion obtained with the Nyström extension, and the minimal-shift embedding.

To obtain a similarity measure to be used with the dominant set approach and the Nyström extension, we combine the two distances with the formula

$$s(p,q) = e^{-\frac{1}{2}\ \frac{d_c(p,q)}{k_1}+\frac{d_f(p,q)}{k_2}}$$

where k_1 and k_2 are scaling constants determined experimentally. On the other hand, the distance used for the minimal shift embedding algorithm is a simple convex combination of the two distances:

$$d(p,q) = \alpha d_c(p,q) + (1-\alpha) d_f(p,q)\,.$$

4.4 Video segmentation

In order to test the performance of the segmentation algorithms, we applied the methods to three video sequences. All the experiments where run on a Pentium 4 PC with with a 1.2GHz CPU and 512Mb of RAM. To ensure a fair comparison, the clustering and extension algorithms where all coded in Matlab, while the feature extraction and distance/similarity calculation where performed using

	Hula (sec.)	Flight (sec.)	Flower garden (sec.)
Dominant set	73	277	279
Nyström extension	871	618*	572*
Minimal-shift embedding	546	1236	1218

* Nyström extension could not handle more than 3 frames.

Table 1. Computation time required to cluster the video sequences.

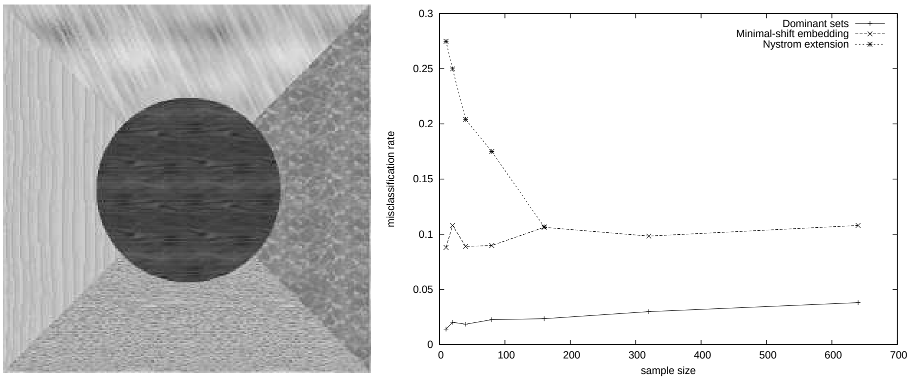


Fig. 6. Sensitivity of the clustering algorithms to the number of samples

non-optimized C++ code. The parameters defining the similarities/distances were experimentally selected to provide the best results, and the number of cluster to be extracted was set to be the same for all three algorithms.

Figures 1–5 show the segmentations obtained from the three sequences, while Table 1 displays the time required to segment the videos.

In Figure 1 we can see the results on the "Hula" sequence. This sequence consists of 8 frames of 160x128 pixels each and 100 samples were used to extract the clusters. Lines 2 to 4 of Figure 1 show the segmentation obtained using the dominant set approach, the Nyström extension and the minimal-shift embedding approach respectively. In the segmentation images every grey-level represent a different cluster. In the video there is a clear separation between figure and ground and all three algorithms provide similar segmentations. Note however, that there is a dramatic difference in computation time, with the dominant set approach clearly outperforming the other two. Figures 2 and 3 display the segmentation obtained from the "flight" sequence. This sequence consists of 12 frames of 160x128 pixels each and 300 samples were used to extract the clusters. It is important to note that, already with this small example, the size of matrix B of the similarities between out-of-sample points and sample points was too big to fit in memory, rendering the Nyström method infeasible. In fact The approach could not be applied to more than 3 frames. The other two approaches, on the other hand, did not suffer from this limitation due to their ability to perform the extension *on-line*. The sequence was much harder to segment, and all three approaches provide rather poor, yet comparable, segmentations. However, in this case as well the dominant set approach clearly outperforms both methods in terms of execution time. Finally, Figures 4 and 5 present the segmentation obtained from the “flower garden” sequence. This sequence consists of 12 frames of 176x120 pixels each. 300 samples were used to extract the clusters, and in this case as well the Nyström method could not go beyond the first 3 frames. In this case the dominant set, not only proved much faster than the other approaches, but also provided a better segmentation than those obtained with the other algorithms. In fact it was the only method that proved to be able to separate the tree in the foreground from the roofs of the houses in the background.

4.5 Sensitivity analysis

To complement these real world experiments we have performed some sensitivity analysis aimed at assessing the performance of the clustering algorithms as the number of samples is reduced. To this end we have applied the segmentation approaches to an artificial image with 5 patches of different texture. The segmentation were repeated using 10, 20, 40, 80, 160, 320, and 640 color and texture samples. Figure 6 displays the image to be segmented and the rate of misclassified points as the number of samples increases. Note that the Nyström approach did not scale beyond 160 samples. The results show that both the dominant set approach and the minimal-shift embedding method are insensitive to the number of samples, with the dominant set approach having the edge in terms of average misclassification rate. The Nyström method, however, shows a clear dependency on the number of samples.

5 Conclusions

In this paper we have addressed the problem of applying the dominant set approach to spatio-temporal segmentation of video feeds. In order to do this we proposed an efficient approach to assigning out-of-sample data to one of a set of previously determined clusters. This allowed us to substantially reduce the computational burden associated to the processing of the huge amount of data involved in video segmentation. We compared the performance of the approach to two state-of-the-art out-of-sample cluster-extension algorithms: the Nyström extension applied to normalized cut and the minimal-shift embedding method. The experiments show that, in terms of quality, the algorithm performs comparably to the minimal shift embedding and the Nyström extension. In terms of computation time, the dominant set approach proved to be consistently the fastest of the three algorithms.

References

1. Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral cluster-

ing. In *Advances in Neural Information Processing Systems 16*, MIT Press, Cambridge, MA, 2004.

2. B. Fischer, J. M. Buhmann. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. Pattern Anal. Machine Intell.* 25(4):513–518, 2003.
3. C. Fowlkes, S. Belongie, F. Chun, and J. Malik. Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Machine Intell.* 26:214–225, 2004.
4. R. Hamid et al. Detection and explanation of anomalous activities: representing activities as bags of event n-grams. To appear *CVPR* 2005.
5. T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Trans. Pattern Anal. Machine Intell.* 19:1–14, 1997.
6. D. Luenberger. *Linear and nonlinear programming.* Addison-Wesley, Reading, MA, 1984.
7. J. Mailk, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. J. of Computer Vision* 43(1):7–27, 2001.
8. T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of Turán. *Canad. J. Math.* 17:533–540, 1965.
9. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, MIT Press, Cambridge, MA, pp. 849–856, 2002.
10. M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 145–152, 2003.
11. M. Pavan, M. Pelillo. Unsupervised texture segmentation by dominant sets and game dynamics. In *Proc. 12th Int. Conf. on Image Analysis and Processing*, pp. 302–307, 2003.
12. M. Pavan and M. Pelillo. Efficient out-of-sample extension of dominant-set clusters. In *Advances in Neural Information Processing Systems 17*, MIT Press, Cambridge, MA, pp. 1057–1064, 2005
13. P. Perona and W. Freeman. A factorization approach to grouping. In: H. Burkhardt and B. Neumann (Eds.), *Computer Vision—ECCV'98*, pp. 655–670. Springer, Berlin, 1998.
14. V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. Pattern Anal. Machine Intell.* 25:1540–1551, 2003.
15. S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding* 71:110–136, 1998.
16. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.* 22:888–905, 2000.
17. J. W. Weibull. *Evolutionary game theory.* MIT Press, Cambridge, MA, 1995.
18. Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proc. 7th Int. Conf. on Computer Vision*, pp. 975–982, 1999.