LAMP-094
CAR-TR-979
CS-TR-4403
UMIACS-2002-83

MDA9049-6C-1250
August 2002

A Survey of Spatio-Temporal Grouping Techniques

Remi Megret[1] and Daniel DeMenthon[2]

[1]Laboratoire RFV
INSA Lyon, Villeurbanne, 69621 CEDEX, France
megret@rfv.insa-lyon.fr
[2]Language and Media Processing (LAMP)
University of Maryland, College Park, MD 20742-3275
daniel@cfar.umd.edu

Abstract

Spatio-temporal segmentation of video sequences attempts to extract backgrounds and independent objects in the dynamic scenes captured in the sequences. It is an essential step of video analysis. It has important applications in video coding, video logging, indexing and retrieval, and more generally in scene interpretation and video understanding. We classify spatio-temporal grouping techniques into three categories: (1) segmentation with spatial priority, (2) segmentation by trajectory grouping, and (3) joint spatial and temporal segmentation. The first category is the broadest, as it inherits the legacy techniques of image segmentation and motion segmentation. The other two categories place a higher priority on the accumulation of evidence along the temporal dimension and are more recent developments made feasible by the increased availability of computing power. For each category we provide a taxonomy of the techniques used to produce meaningful pixel groupings.

The support of this research by the Department of Defense under contract MDA 9049-6C-1250 is gratefully acknowledged.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
AUG 2002		**00-08-2002 to 00-08-2002**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Survey of Spatio-Temporal Grouping Techniques	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Language and Media Processing Laboratory,Institute for Advanced Computer Studies,University of Maryland,College Park,MD,20742-3275**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**16**	

1 Introduction

A natural description of a video is a decomposition into objects. The objects may include semantic entities, or visual structures such as color patches. Segmenting objects automatically is one of the most challenging tasks in video processing, even though human vision seems to achieve it effortlessly.

Such a segmentation has numerous applications, including compact video coding, automatic and semi-automatic content based description, film post-production, and scene interpretation.

Transmission of videos requires a great amount of compression, especially in wireless applications. The ability to extract backgrounds and moving objects makes it possible to eliminate the redundancy related to the repetition of the same visual patterns in successive images.

For video description tasks, such as logging and annotation, automatic object extraction can help in building high level indexes that take into account the fact that a scene is generally composed of multiple entities of interest. A tool able to provide a structured representation and a segmentation into objects is valuable as it facilitates visualization and annotation by a human operator. Such annotations enrich raw video content with object-specific information, which can then be used by search engines and interactive multimedia documents.

Automatic object segmentation is also useful in post-production, when special effects and visual modifications must be independently added to background scenes and foreground action.

Finally, scene interpretation is largely dependant on object extraction. It can be performed automatically in restricted contexts where *a priori* constraints can be enforced, such as sports video understanding and video surveillance.

The broad variety of potential applications is mirrored by an equally broad variety of approaches and specifications. Objects can be defined at several levels. The most familiar level to humans is the semantic level, where each part of an image (and subsequently of a video) is labeled as its counterpart in the real world: a hand, a person, a car. This requires an interpretation of the scene, which is subject to the ambiguity and variations associated with subjective evaluation: two different persons may define objects of interest differently, and thus prefer different segmentations.

Except in restricted domains, the semantic level is generally not computable automatically, since it requires some amount of scene interpretation. Therefore segmentation methods rely on concrete and measurable segmentation criteria that define non-semantic entities. Two main types of methods are used with videos, alone or in combination: motion-based methods, and color/texture-based methods.

Motion based segmentation methods make implicit modeling assumptions about the video capture process, which associates object geometry and displacements in the scene with specific apparent motion in the video. Underlying hypotheses include rigid body motion and spatial smoothness of motion. When these hypotheses are verified, motion segmentation reaches, at least partly, the semantic level, by its ability to extract real objects moving independently.

If these hypotheses are not verified, lower-level structure can still be extracted, based on visual features such as color and texture. As is experienced in image segmentation, these have a poor semantic value, since a visual object may be composed of several distinct colors or, less frequently, separate objects may have similar colors. Nevertheless, they provide a structured representation of raw video data, by extracting from the spatio-temporal volume the pixels spanned by patches of homogeneous visual characteristics. A patch segmentation is useful in itself as a low-level

representation, or as an input for higher level modules, such as motion and event analysis.

2 Overview of Classification

Spatio-temporal grouping manipulates features embedded in the spatio-temporal volume, the *video stack*, produced by the stacking of the individual consecutive video frames. The spatial and temporal dimensions of this volume can be handled either separately or simultaneously.

Most approaches handle these two types of dimensions separately, making a distinction between spatial segmentation, which groups features using spatial coherence criteria, and temporal tracking, which groups features using a temporal invariance hypothesis. The order in which spatial and temporal groupings are performed leads to two different approaches: *segmentation with spatial priority*, and *trajectory grouping*.

Segmentation with spatial priority first focuses on the spatial segmentation of each frame in the video stack. Spatio-temporal groups are then obtained as the extension in time of existing spatial segments. This category includes motion segmentation based on similarity of instantaneous motion [1] [2] [3] [4] and motion model fitting [1] [5] [6], as well as color/texture segmentation and region tracking [7] [8] [9].

On the other hand, trajectory grouping first considers temporal grouping, tracking discrete features to extract their trajectories. Then trajectories belonging to the same moving objects are spatially grouped together using motion segmentation. This family may be divided into methods using motion similarity [10] [11] [12], and methods using explicit motion model fitting [13] [14]. When a dense segmentation is needed, an optional densification step may be added, which fleshes out discrete trajectory features with neighboring sets of pixels.

A more recent class of methods, *joint spatial and temporal grouping*, avoids favoring one dimension over the other and instead operates directly in the spatio-temporal volume. These methods define the grouping criteria simultaneously in space and time, so that evidence for groupings is gathered at the same time in both dimensions. They rely on pixel color and spatio-temporal position [15], or also incorporate instantaneous motion [16] [17] [18].

This previous classification of approaches is represented in Figure 1. Spatio-temporal grouping starts with unstructured features, such as image features (color, texture, motion field...), or discrete features (interest points, edges...). Segmentation with spatial priority first groups spatially, then extends the segmentation temporally. Trajectory grouping tracks features temporally before grouping the resulting trajectories spatially. Finally, joint spatio-temporal grouping follows a diagonal path and builds structures in both dimensions simultaneously.

Section 3 briefly describes the building blocks used by various grouping methods. Category-specific descriptions are then developed in Section 4 (segmentation with spatial priority), Section 5 (trajectory grouping) and Section 6 (grouping jointly in space and time).

3 Building Blocks of the Grouping Process

Before examining each category in detail, it is of interest to compare the building blocks used by various grouping methods. These are either individual image pixels [1] [5] [19] [20], spatial regions resulting from a color or texture over-segmentation [21] [3] [2] [4], or discrete geometric features such as interest points [22] [13] [23] [24] and edges [13].

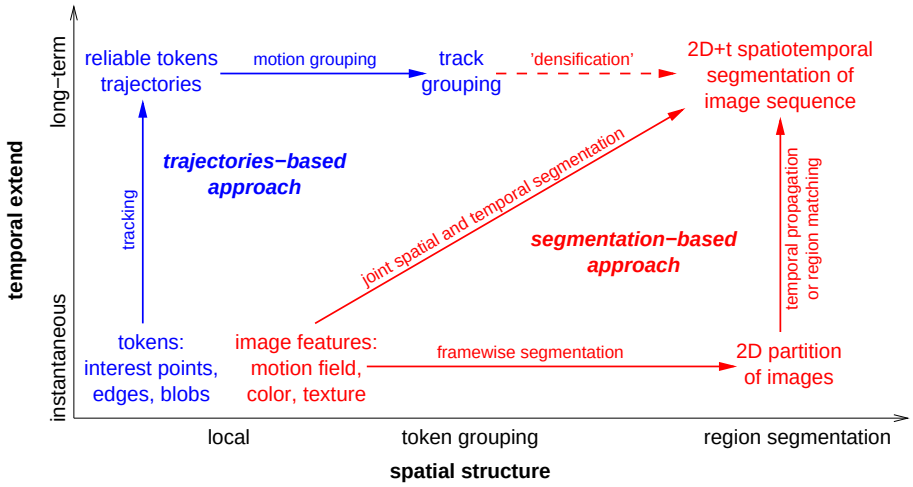

Figure 1: Structuration flows along temporal and spatial axes.

Pixel and region methods have the advantage of working directly on the original data, which implicitly provides a topology for regularization, as well as color and texture features. Discrete features have been mostly used in pure motion segmentation frameworks, where they provide an explicit representation of displacements. Trajectory grouping is based on discrete features, since these can be tracked over an extended time interval.

Region methods, in contrast to pixel methods, manipulate a smaller number of features, thus reducing complexity. The extended spatial support also makes the initial estimation of motion more accurate and robust. However, color and texture alone may not be sufficient to over-segment expected object boundaries.

Note that some methods may consider different kinds of building blocks at different stages of the process. For example, Wang and Adelson [1] first merge a priori regions based on motion parameters in the initialization step, but they later come back to the pixel level when refining motion and spatial supports of layers.

4 Segmentation with Spatial Priority

The first category gives priority to spatial segmentation of image features. Methods in this category can be seen as an extension of single frame segmentation by adding temporal tracking.

We further divide these methods into methods using motion segmentation, and methods using color/texture segmentation. Methods relying on motion are mostly sequential. They segment frames one after the other based on instantaneous motion and segmentation in the previous frame. Methods based on color/texture can also segment video frames individually, then merge the regions temporally.

4.1 Sequential Motion Segmentation

Spatio-temporal segmentation based on instantaneous motion is a very broad family of methods that we analyze with respect to the two steps it involves: the method used for framewise motion segmentation, and how temporal coherence is enforced.

4.1.1 Framewise Motion Segmentation

Whether the building blocks are pixels, regions, or discrete features, motion segmentation involves two kinds of techniques: motion similarity techniques and model fitting techniques. We note that both approaches rely on an underlying motion model, which may be either a spatial motion smoothness model or a parametric (translational, affine, perspective) model. The distinction we propose emphasizes how this model is applied to the video data.

- *Motion similarity* methods estimate motion parameters on a local basis, for each element independently, or for each pair of elements. The grouping involves a symetrical comparison between elements of the same nature, for example while clustering in motion parameter space, or grouping pairs of similar elements.
- *Motion model fitting* methods compute motion parameters in groups of identically labeled elements. They involve the evaluation of asymmetric measures of the quality of fit of an element to a motion model.

Figure 2 represents in a schematic way a taxonomy of techniques used in segmentation with spatial priority. Detailed explanations can be found in the corresponding paragraphs of this section.

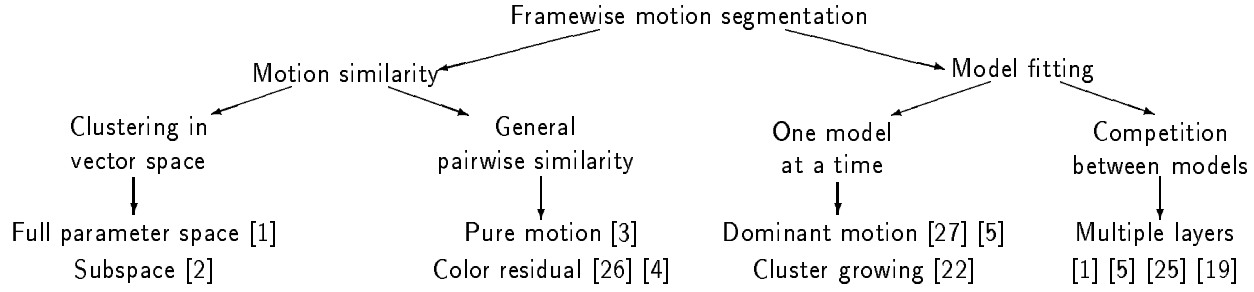

Figure 2: Taxonomy of grouping methods with spatial priority from the point of view of spatial segmentation and temporal coherence.

Motion Similarity Segmentation

Motion Parameter Space The simplest measure of motion similarity is obtained by using Euclidean distance in motion parameter space. To initialize their motion models, Wang and Adelson [1] compute affine motion on *a priori* regions, then cluster them in parameter space using this measure.

In the work of Ke and Kanade [2], the same clustering approach is considered, and is further constrained by first projecting on a subspace of the full parameter space. The resulting dimensionality reduction contributes to decreasing the disparity inside clusters. This reduction cannot be dramatic, though, since it starts from a six-dimensional space. The authors suggest, as future work, taking into account motion parameters computed on successive frames, so that the full parameter space has more dimensions. In that context they could more effectively take advantage of

the claimed property that motion parameters of rigid objects lie in a three-dimensional subspace, independently of the number of frames considered.

Affine motion with parameters $(a_{x0}\ a_{xx}\ a_{xy}\ a_{y0}\ a_{yx}\ a_{yy})$ leads to the motion vector $(u,v)^{\mathrm{T}}$ at a given position $(x,y)^{\mathrm{T}}$:

$$\left\{ \begin{array}{lcl} u & = & a_{x0} + x\, a_{xx} + y\, a_{xy} \\ v & = & a_{y0} + x\, a_{yx} + y\, a_{yy} \end{array} \right. \tag{1}$$

The six parameters can be separated into two sets with different homogeneities: two for zero-order (translation: $(a_{x0}\ a_{y0})$) and four for first-order (rotation, zoom, shear: $(a_{xx}\ a_{xy}\ a_{yx}\ a_{yy})$). The sensitivity of predicted motion vectors to a zero-order parameter is uniform, whereas its sensitivity to a first-order parameter depends on spatial position, and is higher for positions far from the origin. To make sure that all dimensions have roughly the same influence on the borders of the image, Wang and Adelson [1] normalize the parameters by dividing the first-order parameters by the size of the image.

Pairwise Contextual Similarities To avoid the problem of interpretation and normalization of metrics in motion parameter space, other methods avoid comparing motion parameters directly, but rather come back to the spatial domain, where motion similarity can be expressed in terms of physical values: motion vector discrepancy [3], or error of motion-compensated pixel values [26] [4].

Gelgon and Bouthemy [3] compute affine motion parameters for each individual region. For a pair of neighboring regions, they compare the motion fields predicted by the two parametric models on the union of the regions. This motion similarity reflects how well the motion associated with each region predicts the motion on the other. They incorporate these motion similarities in a Markov Random Field (MRF) segmentation framework.

Such a cross-validation of motion models is also used for the merging of spatial regions by Moscheni et al. [26]. They base the similarity on the residuals of motion-compensated pixel values between successive frames. This criterion is combined with a contrast criterion, to favor grouping of regions of same luminance. They use the same framework to link regions temporally, by defining a similarity based on invariance.

Residuals are used in the region merging step of Wang [4]; he associates with every pair of adjacent regions the motion model computed on their union. He merges the two regions if the compensation error on pixel values is below a given threshold.

Model Fitting Model fitting methods are based on the notion of quality of fit of each element to the model. These methods seek to find the model parameters that will optimize the overall quality of fitting.

Top-down approaches consider a reference motion model, and classify features into inliers and outliers. The inliers are usually associated with the background, while the outliers correspond to the foreground objects. The reference motion, which usually corresponds to the motion of the dominant object, is estimated using robust methods, which are less sensitive to other motions and can be applied to the common occurrence of a background that has a parametric motion and is the dominant object [5]. Several objects can be extracted by recursively applying this method to remaining outliers [27].

Growing clusters in a bottom-up manner, Smith and Brady [22] group motion vectors associated with interest points. Points are added one by one to existing clusters, by testing if their motion vectors are close enough to the vector predicted at the same point by the affine motion model of the cluster.

Approaches related to a *layered representation* [1] [5] [25] take into account several models that compete with each other. This can be expressed in a probabilistic framework by *mixture models*: each feature (pixel or region) is associated with one motion model; the parameters of the models are unknown.

Joint estimation of both motion parameters and labels is very complex and prone to being trapped in local minima, because of its very high dimensionality (labeling has one variable per element). This problem is generally solved using the Expectation-Maximization (EM) framework, which estimates one of the unknowns, while keeping the others constant, and iterates until convergence.

Most work of this type associate an affine parametric model with each layer [5] [1]. In [19] only motion smoothness inside each layer is enforced.

4.1.2 Spatial Segmentation using Color/Texture

Some methods base the segmentation on visual features such as color or texture. This is seen either as a way of finding a spatio-temporal representation based on space-time tubes of consistent color/texture, or as a preprocessing step before motion segmentation.

With noisy data, or in the presence of non-rigid objects, motion fields may be unreliable or non-parametric. In such cases, it is possible to rely entirely on color or texture, and then group segmented spatial regions temporally. Deng and Manjunath [7] use this approach (see below for details); Wang [4] and Gomila [28] use morphological color segmentation; Del Bimbo et al. [29] find clusters in color feature space.

This approach usually attempts to over-segment objects in each frame, and then to match the pieces from frame to frame; this works better for objects that have parts with uniform color/texture, and high contrast with other objects.

In Deng and Manjunath's work, [7], seeds derived from a segmentation of the previous frame are projected to the current frame assuming slow motion, and grouped with those segmented regions of the current frame that overlap them. Del Bimbo et al. [29] compute similarities between regions in two successive frames based on color invariance, and spatial overlapping. They connect each region to the best match in the next frame, unless similarity is under a threshold.

Faster motions can be handled by considering the overlap of motion-compensated regions, as in the work of Wang [4]. In case of conflict, he favors the temporal grouping between pairs of regions that have the smallest difference in pixel values.

In [8], the matching is not computed on successive frames, but between successive groups of frames.

4.1.3 Temporal Coherence

Temporal grouping is an important step of spatio-temporal segmentation which deserves futher comparative discussion. The result should enforce identical labels for the same object across frames.

Most of the techniques based on spatial motion segmentation handle frames sequentially. They project the segmentation computed in one frame onto the next, and take it into account while segmenting the new frame. This is justified by the causal relationship between frames, and is attractive because frames can then be treated in the order in which they are decoded or acquired. On the other hand, this doesn't use all available information; indeed, only framewise motion information is available at the beginning of the sequence, and most of the time, information accumulated over a sequence is reduced to the spatial segmentation of the previous frame.

Temporal coherence can be enforced by two kinds of techniques: initialization from the previous frame and explicit temporal constraints.

Initialization from the previous frame is used in iterative methods, where the final solution is found by successive changes to an initial labeling. The segmentation in the current frame is initialized according to a prediction computed by projecting the previous segmentation [1] [5] [21] [3] [4] [22]. The iterations then converge to a local optimum, given the high dimensionality of the solution space, which lies in a valley dependent on the initialization. During the optimization, the final result may drift away from the initialization during the iterative optimization [9](p112). This problem arises mainly in the presence of segmentation ambiguities, where the optimum may vary from one frame to the other; for example, when a moving object is almost still, the motion boundaries may be difficult to extract accurately in a single frame.

To enforce stronger temporal contraints, some methods combine the framewise segmentation criterion with an additional term, which explicitly models the temporal coherence of labelings. When segmentation is ambiguous at the single-frame level, temporal constraints ensure a better coherence. This coherence can be expressed as the invariance of the labels with respect to the projection from the previous frame [30] [21], or as the fitting with a location model that depends on several previous frames [31].

Patras et al. [21] use a MRF framework in which the Gibbs potential of a single site (region) takes into account the number of pixels for which this region has the same label as the motion-compensated projection of the labels from the previous frame.

The condition of temporal order is relaxed by Jojic and Frey [25]. They use a mixture model with four types of hidden variables: the appearance model of each class (called a sprite) and its variance, the actual appearances of each sprite in each frame of the sequence, the spatial transformations that map these appearances onto each frame, and finally the masks that tell which pixels of the sprites are seen in each frame.

More restrictive and longer-term constraints are introduced by Sawhney and Kumar [31]. They impose temporal constraints that penalize changes in motion and in segmentation shape over several successive frames.

5 Trajectory Grouping

Methods with spatial priority in the motion segmentation category described above rely only on short-term motion information (usually between two frames). To take into account long-term information, another class of methods has been developed using trajectories that can represent the motions of points in a long temporal interval. In this case, less ambiguous displacement differences can be observed, and motions are better discriminated.

The estimation of trajectories is performed as a preliminary step, using feature point temporal

matching [32], or textured patch tracking [33]. A drawback of this approach is that since the spatial motion segmentation takes place afterward, tracking cannot use any *a priori* spatial constraints, such as parametric motion or group tracking, which would improve quality and efficiency. For this reason, this approach is best suited to applications where tracking can be performed reliably enough to produce trajectories almost free of noise. This is the case in sequences with slow motion, and when there are enough discriminating features on the objects of interest.

The next sections describe the main trajectory grouping techniques, which are summarized in figure 3.

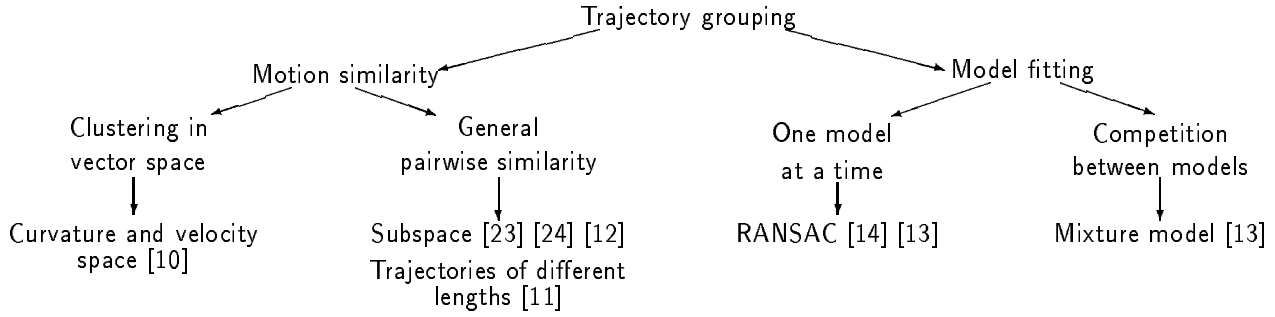

Figure 3: Taxonomy of trajectory grouping methods.

5.1 Grouping by Motion Similarity

5.1.1 Direct Comparison of Trajectories

Methods based on direct comparision of trajectories define a similarity between two trajectories which is not influenced by the other trajectories. It can consist in representing each trajectory as a point in a multidimensionnal vector space and then use Euclidean distances as in [10] , or define a more general pairwise motion similarity as in [11] and [34].

Allmen and Dyer [10] compute trajectories, which they call spatio-temporal flow curves, by integrating local motion flow over time. Given a fixed-width temporal interval, they represent the trajectories by their curvature and slope values in that interval. They cluster those features using a K-means algorithm, which groups together trajectories having similar curvatures and slopes. The analysis of the merging and splitting of clusters as the time interval is shifted gives information about occlusion events.

Megret and Jolion [11] propose a hierarchical clustering framework that allows them to use tracks of different lengths. Track similarity is based on the invariance of relative position in several frames. Hierarchical agglomerative clustering is then used to produce a clustering tree that groups together similar long-term motions.

Mills and Novins [34] consider 3D spatial feature points. They create a *feature interval graph* that associates with each pair of feature points in the scene an interval that represents the possible values of their 3D distance. The intersection of the intervals obtained from two tracked features points at different times then reveals whether they belong to the same rigid object.

5.1.2 Subspace Factorization

Subspace methods represent a trajectory as the vector of the coordinates of its feature points over time, and stack them in a matrix C. With an affine camera, the tracks associated with differently moving rigid bodies moving differently lie in separate subspaces. The core of the method is to reduce C to a form that enhances these subspaces.

Costeira and Kanade [23] and Gear [24] factorize the matrix C using singular value decomposition (SVD): $C = U\Sigma V^{\mathrm{T}}$, where U and V are orthonormal matrices, and Σ is diagonal. This decomposition is then truncated to the rank k of C, by keeping only highest singular values: $C_k = U_k \Sigma_k {V_k}^{\mathrm{T}}$. This results in a *shape interaction* matrix $Q = V_k {V_k}^{\mathrm{T}}$, which has the following property: the interaction coefficient Q_{ij} is zero if trajectories i and j belong to separate rigid objects, and is nonzero otherwise.

The final segmentation is computed by clustering the points representing trajectories, based on the values in the matrix Q. Such a clustering can also be seen as block diagonalization of Q. Costeira and Kanade [23] use a greedy algorithm, which recursively merges groups of trajectories having high interactions. This matrix is also used by Ichimura [35], who uses a recursive subdivisivion approach guided by discriminant analysis.

An interesting interpretation of methods based on the matrix Q was proposed by Weiss [6], who shows that they amount to an eigen-clustering using an affinity matrix W produced by the inner product of trajectory coordinates:

$$W_{ij} = \sum_t x_{ti} x_{tj} + y_{ti} y_{tj} \tag{2}$$

In order to decrease the influence of noise, Kanatani [12] and Zhang et al. [36] fit subspaces to groups of points, instead of relying only on pairwise affinity. Kanatani [12] uses a merging approach where points that belong to a group are projected onto a subspace fitted to the group. The merging decision also takes into account model selection. Zhang et al. [36] use Q as a preprocessing step to produce an over-segmentation, then estimates group distances D between these segments. Final object segmentation is produced by thresholding on D.

In real data, trajectories may have missing points in some frames, which precludes the direct use of SVD. Tomasi and Kanade [37] apply SVD only to fully defined vectors. Jacobs [38] and Shum et al. [39] take into account all available data, by fitting a low-rank matrix to the incomplete data matrix. The fitting may be an iterative weighted least-square method [39], or a direct method based on constraints derived from a set of fully defined submatrices [38].

5.2 Grouping using Explicit Parametric Models

5.2.1 Hypothesize and Test

Hypothesize and test methods work as follows: hypotheses are obtained by fitting models to small data point sets chosen randomly. Each hypothesis is then validated by assessing the quality of fit. In RANSAC based methods, this is achieved by counting the number of inlier points. Hypotheses that have enough inliers are kept, and possibly compared to each other in order to merge similar ones.

This method copes well with outliers. Indeed, by choosing small sets of points, one increases the possibility of considering only points of the same model, thus computing a correct parameter

vector. This is not the case when all the data is used simultaneously, since outliers are then included in the estimate. The threshold must be fixed in advance to decide how many points are actually inliers. Setting it requires some preknowledge about the expected error around the model.

Baldi et al. [14] apply an affine motion model to cluster trajectories for mosaicking. They use a variant where point sets are not chosen randomly, but deterministically by considering only spatial neighborhoods. Torr and Zisserman [13] refer to such sets as propinquintal sets. Choosing only sets of neighboring points increases the chance of picking points in the same object. This avoids, for example, incorrectly assigning a single rotational model to two translating groups of features [14].

5.2.2 Motion Mixture Models

Torr and Zisserman [13] adopt a mixture model formulation, which associates each trajectory with an object model; each object model consists of a parametric motion model, which describes the displacement of each point in the image over the whole sequence. Estimation of labels of trajectories (linking each trajectory to an object model) and motion parameter estimation are performed using an EM approach. The initialization comes from RANSAC. Although examples on only three frames are provided, the formulation allows for the use of longer trajectories, as long as they are defined on the same temporal interval.

6 Joint Spatial and Temporal Segmentation

In contrast to other methods, which give priority to spatial or temporal grouping, joint spatial and temporal segmentation methods consider a video as a spatio-temporal block of pixels, by treating the spatial and temporal dimensions simultaneously. The merit of this approach is supported by Gepshtein and Kubovy [40], who suggest that human vision finds salient structures jointly in space and time.

Figure 4 summarizes the different approaches used for joint spatial and temporal grouping techniques. We detail them in the following subsections. First we explore grouping in a vector space, either using similarity clustering [16], or fitting of a mixture-model [15]. In a second subsection, we present methods based on graph cut [17] [18].

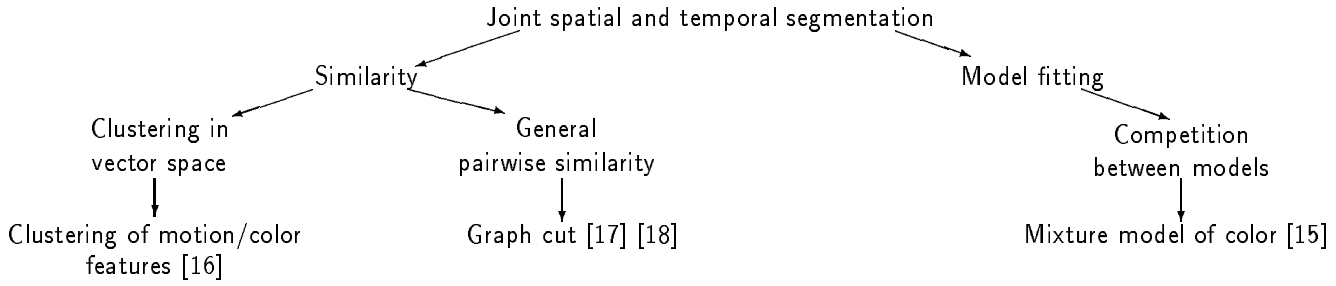

Figure 4: Taxonomy of joint spatial and temporal grouping methods.

6.1 Clustering in Feature Space

Greenspan et al. [15] consider videos in a six-dimensional feature space: color, spatial and temporal position. Each pixel is associated with a point in the feature space. Pixels are clustered using a Gaussian mixture model, in an EM framework, where the number of clusters is fixed a priori. The optimal number of clusters is found using MDL, which requires running the clustering several time with different number of clusters. Once the model is learned, Gaussian covariance coefficients between spatial and temporal dimensions give information about motion, which is used for event detection. A hard labeling of pixels can be found by assigning each pixel to the Gaussian distribution that best explains its value.

DeMenthon [16] precomputes an optical flow at each pixel, whose spatio-temporal orientation and position are respectively represented by two motion angles and two motion distances that are invariant to the motion shifts of the pixel. The seven-dimensional feature vectors composed of three color and four motion descriptors have the property that pixels having a similar color and belonging to a linearly moving moving patch are close to each other in feature space. They are clustered using a hierarchical method derived from mean shift.

6.2 Graph-Based Segmentation

These grouping techniques are the extension to the spatio-temporal volume of graph-based image segmentation [41, 42, 6]. Graph-based methods consider a graph whose nodes are the image features (pixels taken from the whole video volume), and whose edges are weighted according to some measure of similarity between nodes (also called *affinity* in this context). The nodes are grouped using graph cut techniques. The edges connect pixels in spatial as well as temporal directions, thus yielding a joint spatial and temporal segmentation.

Shi and Malik [17] use a similarity based on motion profiles. A motion profile represents the probability distribution of the motion vector at a given point.

Fowlkes et al. [18] use a similarity based on several visual cues. They attach to each pixel i of the sequence a feature vector $\mathbf{x}_i$ containing its spatio-temporal location (x, y, t), its color in (L, a, b) space, and optical flow (u, v). The affinity between pixels i and j is defined as

$$W_{ij} = \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}_i - \mathbf{x}_j)\right\} \tag{3}$$

where Σ gives the weights of the dimensions. They also use the Nyström approximation of the normalized cut algorithm to compute segmentations more efficiently. This method approximates the similarity matrix by sampling the input features and expressing all pairwise similarities with respect to those samples. The resulting speedup is necessary because of the large number of pixels in the video stack and the complexity of eigenvalue analysis.

7 Concluding Remarks

In this paper, we have surveyed methods of spatio-temporal grouping in videos. The grouping involves building blocks that can be pixels, regions resulting from oversegmentation, or discrete features. These blocks are grouped based on similar motion, or similar color/texture. We propose

a classification which has three categories: (1) segmentation with spatial priority, (2) segmentation by trajectory grouping, and (3) joint spatial and temporal segmentation.

Inside each category, we also make a distinction between methods based on pairwise similarities, which lead to simple grouping criteria and can give global optima, and model fitting methods, which can represent more complex criteria, at the expense of iterative computation, and additional a priori knowledge.

This study has been limited to techniques which tackle the problem of object extraction in monocular videos for low-level features. We didn't address object detection methods devoted to special applications, for which more information on objects is available, because of specialized knowledge on the nature of the objects or tracking initialization. In particular, readers may refer to methods developped for tracking individual objects [43] or for the detection and the recognition of humans activities [44].

References

[1] J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.

[2] Q. Ke and T. Kanade, "A subspace approach to layer extraction," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 255–262, 2001.

[3] M. Gelgon and P. Bouthemy, "A region-level motion-based graph representation and labeling for tracking a spatial image partition," *Pattern Recognition*, vol. 33, no. 4, pp. 725–740, 2000.

[4] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 539–546, 1998.

[5] H. S. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 814–830, 1996.

[6] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *IEEE International Conference on Computer Vision*, pp. 975–982, 1999.

[7] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[8] Y. Deng and B. Manjunath, "NeTra-V: Toward an object based video representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 616–627, 1998.

[9] R. Castagno, *Video segmentation based on multiple features for interactive and automatic multimedia applications*. thèse, EPFL, 1999.

[10] M. Allmen and C. R. Dyer, "Computing spatiotemporal relations for dynamic perceptual organization," *CVGIP: Image Understanding*, vol. 3, no. 58, pp. 338–351, 1993.

[11] R. Megret and J.-M. Jolion, "Grey-level blobs tracking for video dynamic content representation," in *Reconnaissance de Formes et Intelligence Artificielle*, vol. 2, pp. 397–406, 2002. (in french).

[12] K. Kanatani, "Motion segmentation by subspace separation: Model selection and reliability evaluation," *International Journal of Image and Graphics*, vol. 2, no. 2, pp. 179–197, 2002.

[13] P. H. S. Torr and A. Zisserman, "Concerning Bayesian motion segmentation, model averaging, matching and the trifocal tensor," in *European Conference on Computer Vision*, vol. 1, pp. 511–527, LNCS 1406, Springer-Verlag, Berlin, Germany, 1998.

[14] G. Baldi, C. Colombo, and A. Del Bimbo, "A compact and retrieval-oriented Video representation using mosaics," in *International Conference on Visual Information Systems (VISUAL)*, pp. 171–178, LNCS 1614, Springer-Verlag, Berlin, Germany, 1999.

[15] H. Greenspan, J. Goldberger, and A. Mayer, "A probabilistic framework for spatio-temporal video representation and indexing," in *European Conference on Computer Vision*, vol. 4, pp. 461–475, LNCS 2353, Springer-Verlag, Berlin, Germany, 2002.

[16] D. DeMenthon, "Spatio-temporal segmentation of video by hierarchical mean shift analysis," *Statistical Methods in Video Processing Workshop*, 2002.

[17] J. Shi and J. Malik, "Motion segmentation and tracking using Normalized Cuts," in *IEEE International Conference on Computer Vision*, pp. 1151–1160, 1998.

[18] C. Fowlkes, S. Belongie, and J. Malik, "Efficient spatiotemporal grouping using the Nystrom method," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 231–238, 2001.

[19] Y. Weiss, "Smoothness in layers: motion segmentation using nonparametric mixture estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 520–526, 1997.

[20] D. Elias and N. Kingsbury, "Image sequence segmentation using mixture models with uniqueness and spatio-temporal consistency constraints," in *IEE Colloquium on Motion Analysis and Tracking*, (London, UK), pp. 6/1–6, 1999.

[21] I. Patras, E. A. Hendriks, and R. L. Lagendijk, "Video segmentation by MAP labeling of watershed segments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 326–332, 2001.

[22] S. M. Smith and J. M. Brady, "ASSET-2: Real-time motion segmentation and object tracking," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 814–820, 1995.

[23] J. Costeira and T. Kanade, "A multi-body factorization method for motion analysis," in *IEEE International Conference on Computer Vision*, pp. 1071–1076, 1995.

[24] C. Gear, "Feature grouping in moving objects," in *IEEE Workshop on motion of non-rigid and articulated objects*, pp. 214–219, 1994.

[25] N. Jojic and B. Frey, "Learning flexible sprites in video layers," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 199–206, 2001.

[26] F. Moscheni, S. Bhattacharjee, and M. Kunt, "Spatiotemporal segmentation based on region merging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 9, pp. 897–915, 1998.

[27] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 5–16, 1994.

[28] C. Gomila, *Mise en correspondance de partitions en vue du suivi d'objets*. PhD thesis, École Nationale Supérieure des Mines de Paris, 2001.

[29] A. Del Bimbo, P. Pala, and L. Tanganelli, "Video retrieval based on dynamics of color flows," in *International Conference on Pattern Recognition*, vol. 1, pp. 851–854, 2000.

[30] N. Brady and N. O. Connor, "Object detection and tracking using an EM-based motion estimation and segmentation framework," in *IEEE International Conference on Image Processing*, vol. 1, pp. 925–928, 1996.

[31] H. Tao, H. S. Sawhney, and R. Kumar, "Object tracking with Bayesian estimation of dynamic layer representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75–89, 2002.

[32] J. Verestoy and D. Chetverikov, "Experimental comparative evaluation of feature point tracking algorithms," in *Evaluation and Validation of Computer Vision Algorithms*, Kluwer Series in Computational Imaging and Vision, pp. 183–194, 2000.

[33] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[34] S. Mills and K. Novins, "Motion segmentation in long image sequences," in *Proceedings of the 11th British Machine Vision Conference*, vol. 1, pp. 162–171, 2000.

[35] N. Ichimura, "Motion segmentation based on factorization method and discriminant criterion," in *IEEE International Conference on Computer Vision*, pp. 600–605, 1999.

[36] Y. Wu, Z. Zhang, T. S. Huang, and J. Y. Lin, "Multibody grouping via orthogonal subspace decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 252–257, 2001.

[37] C. Tomasi and T. Kanade, "Shape and motion for image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[38] D. Jacobs, "Linear fitting with missing data: applications to structure-from-motion and to characterizing intensity images," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 206–212, 1997.

[39] H.-Y. Shum, K. Ikeuchi, and R. Reddy, "Principal component analysis with missing data and its application to polyhedral object modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 854–867, 1995.

[40] S. Gepshtein and M. Kubovy, "The emergence of visual objects in space-time," *Proceedings of the National Academy of Sciences, USA*, vol. 97, no. 14, pp. 8186–8191, 2000.

[41] J. Shi and J. Malik, "Normalized Cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[42] P. Perona and W. Freeman, "A factorization approach to grouping," in *European Conference on Computer Vision*, vol. 1, pp. 655–670, LNCS 1406, Springer-Verlag, Berlin, Germany, 1998.

[43] C. Rasmussen and G. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 560–576, 2001.

[44] D. M. Gavrila, "The visual analysis of human movement: a survey," *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.