Temporal Factorization

Lihi Zelnik-Manor
California Institute of Technology
Pasadena CA, USA
lihi@vision.caltech.edu

Michal Irani
Weizmann Institute of Science
Rehovot, Israel
michal.irani@weizmann.ac.il

http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html

Abstract

The traditional subspace-based approaches to segmentation (often referred to as multi-body factorization approaches) provide spatial clustering/segmentation by grouping together points *moving with consistent* motions. *We are exploring a dual approach to factorization, i.e., obtaining temporal clustering/segmentation by grouping together* frames *capturing consistent* shapes. *Temporal cuts are thus detected at non-rigid changes in the shape of the scene/object. In addition it provides a clustering of frames with consistent shape (but not necessarily same motion). For example, in a sequence showing a face which appears serious at some frames, and is smiling in other frames, all the "serious expression" frames will be grouped together and separated from all the "smile" frames which will be classified as a second group, even though the head may meanwhile undergo various random motions.*

1 Introduction

The traditional subspace-based approaches to multi-body segmentation (e.g., [6, 7, 9]) provide spatial clustering/segmentation by grouping *points* moving with consistent *motions*. This is done by grouping *columns* of the correspondence matrix of [17] (we review the definition in Section 1.1).

In this work we show that to obtain *temporal* grouping of frames we cluster the *rows* of the same correspondence matrix instead of its columns. We show that this provides grouping of *frames* capturing consistent *shapes*, but not necessarily same motion. We further show that, to obtain such shape-based clustering of frames we need *not* develop any new segmentation/clustering scheme. We can use any of the existing algorithms suggested for clustering points (e.g., [6, 7, 9]), but, instead of applying them to the correspondence matrix as is, we apply them to its transpose.

Note, that spatial "multi-body factorization" [6, 7, 9] usually provides a highly sparse segmentation since commonly the number of points which can be tracked reliably along the sequence is low. Dense spatial segmentation requires dense optical flow estimation (e.g., [11]). In contrast, a small number of tracked points suffices to obtain a dense temporal clustering of frames, i.e., a classification of *all* the frames in the video clip. Furthermore, the dimensionality of the data, which is one of the major difficulties in spatial multi-body factorization, is significantly smaller for temporal segmentation. To obtain dense *spatial* factorization of the entire image (e.g., [11]), the number of points equals the number of pixels in the image, which can be extremely large (hundreds of thousands of pixels). This is *not* the case with *temporal* factorization. The number of frames in the video clip is usually only tens or hundreds of frames, and therefore the temporal factorization is *not* time consuming.

The standard approaches to temporal segmentation cut the video sequence into "scenes" or "shots", mainly by drastic changes in image appearance (e.g., [22, 16, 12]). Other approaches are behavior based (e.g., [21, 15]) and segment the video into sub-sequences capturing different events or actions. The approach suggested here is fundamentally different and provides a temporal segmentation and clustering of frames which is based on non-rigid changes in shape. For example, in a sequence showing a face at some frames serious and in other frames smiling, all the "serious expression" frames will be grouped together and separated from all the "smile" frames which will be classified as a second group, even though the head may meanwhile undergo various random motions.

Our way of formulating the problem provides a unified framework for analyzing and comparing a number of previously developed independent methods. This new view of previous work is described in Section 6. For example, we show that the technique of Rui & Anandan [15] can be reformulated in terms of the factorization approach. Our analysis illustrates that their approach will detect cuts at large changes in motion, whereas we detect cuts at non-rigid shape changes. In a different work, Rao & Shah [14] suggested a view-invariant recognition method for complex hand movements. In Section 6 we show that the similarity constraint they use for *matching* shapes is equivalent to the one we use for *separating* between shapes.

We start by defining notation and reviewing the background to the multi-body factorization approach in Section 1.1. In Section 2 we present our approach to shape based temporal factorization and in Section 3 we discuss its physical meaning. This is extended in Section 4 to the case of multiple sequences. Section 5 explores the similarities and differences between the suggested approach and the standard spatial factorization of motion. Lastly, we review some related work in Section 6 and summarize in Section 7.

1.1 Background on Factorization Methods

Let $I_1, \ldots, I_F$ denote a sequence of F frames with N points tracked along the sequence. Let (x_i^f, y_i^f) denote the coordinates of pixel (x_i, y_i) in frame I_f $(i = 1, \ldots, N\ ,\ f = 1, \ldots, F)$. Let X and Y denote two $F \times N$ matrices constructed from the image coordinates of all the points across all frames:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ & & \vdots & \\ x_1^F & x_2^F & \cdots & x_N^F \end{bmatrix} \qquad Y = \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_N^1 \\ y_1^2 & y_2^2 & \cdots & y_N^2 \\ & & \vdots & \\ y_1^F & y_2^F & \cdots & y_N^F \end{bmatrix} \tag{1}$$

Each row in these matrices corresponds to a single frame, and each column corresponds to a single point. Stacking the matrices X and Y of Eq. (1) vertically results in a $2F \times N$ "correspondence matrix" $W = \left[\frac{X}{Y}\right]$. It has been previously shown that under various camera and scene models [17, 8, 5] the correspondence matrix W of a single object can be factorized into motion and shape matrices: $W = MS$ (where M and S are low dimensional). When the scene contains multiple objects (see [6, 7]) we still obtain a factorization into motion and shape matrices $W = MS$, where M is a matrix containing the motions of all objects and S is a block-diagonal matrix containing the shape information of all objects.

2 Temporal Factorization

The traditional subspace-based approaches to multi-body segmentation (e.g., [6, 7, 9]) provide *spatial* clustering of image points by grouping *columns* of the correspondence matrix $W = MS$. Note, that in the correspondence matrix W every column corresponds to a point and every row corresponds to a frame. Thus, to obtain *temporal* clustering of frames we will apply clustering to the *rows* of W instead of its columns. In this section we present the theory behind temporal clustering of frames and suggest methods for obtaining it.

When factoring the correspondence matrix W into motion and shape, the columns of the motion matrix M span the columns of W and the rows of the shape matrix S span the rows of W. Hence, clustering the *columns* of W into independent linear subspaces will group together points which share the same *motion*. Equivalently, clustering the *rows* of the correspondence matrix W will group frames which share the same *shape*. Luckily, to obtain such row-based segmentation/clustering we need *not* develop any new segmentation/clustering scheme. We can use any of the existing algorithms suggested for segmenting/clustering columns (e.g., [6, 7, 9]), but, instead of applying them to W, we will apply them to W^T. We next show why this is true.

When the scene contains multiple (K) objects moving with independent motions, and the columns of W are sorted according to objects, then as was shown in [6] the resulting shape matrix has a block diagonal structure:

$$W = [W_1, \dots, W_K] = [M_1, \dots, M_K] \begin{bmatrix} S_1 & & 0 \\ & \ddots & \\ 0 & & S_K \end{bmatrix} \tag{2}$$

where $W_i = M_i S_i$ is the correspondence matrix of the i-th object, with motion M_i and shape S_i. The correct permutation and grouping of columns of W into $W_1, \dots, W_K$ to obtain the desired separation into independently moving objects was accordingly recovered [6, 7] by seeking a block-diagonal structure for the shape matrix S. In other words, to obtain spatial segmentation of points one can group the columns of W into independent linear subspaces by assuming that W can be factored into a product of two matrices, where the matrix on the right has a block diagonal form.

Now, taking the dual approach: When the sequence includes non-rigid shape changes (Q independent shapes) and the *rows* of W are sorted according to *shape*, then the resulting *motion* matrix has a block diagonal structure:

$$W = \begin{bmatrix} \tilde{W}_1 \\ \vdots \\ \tilde{W}_Q \end{bmatrix} = \begin{bmatrix} \tilde{M}_1 & & 0 \\ & \ddots & \\ 0 & & \tilde{M}_Q \end{bmatrix} \begin{bmatrix} \tilde{S}_1 \\ \vdots \\ \tilde{S}_Q \end{bmatrix} \tag{3}$$

The permutation and grouping of *rows* of W into $\tilde{W}_1, \dots, \tilde{W}_Q$ to obtain the desired separation into frames capturing independent *shapes* can therefore be obtained by seeking a block-diagonal structure for the *motion matrix* M. Temporal factorization, thus, provides a way to classify video frames according to non-rigid motion (which is captured by changes in shape), while being invariant to the rigid motions of both objects and camera.

Note, however, that if we now take the transpose of W we get:

$$W^T = [\tilde{W}_1^T, \ldots, \tilde{W}_Q^T] = [\tilde{S}_1^T, \ldots, \tilde{S}_Q^T] \begin{bmatrix} \tilde{M}_1^T & & 0 \\ & \ddots & \\ 0 & & \tilde{M}_Q^T \end{bmatrix} \tag{4}$$

That is, the matrix W^T can be factored into a product of two matrices where the matrix on the right is block diagonal. This is equivalent to the assumption made in the factorization of W to obtain column clustering. Thus, we can use any of the algorithms suggested for segmenting/clustering columns (e.g., [6, 7, 9]), however, instead of applying them to W we will apply them to W^T.

The common approaches to multi-body factorization segmented the columns of the correspondence matrix $W = \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$. In this matrix there is a single column corresponding to each point and two rows corresponding to each frame. It has been previously shown [8, 18, 11] that the horizontally stacked matrix $W_h = [X, Y]$ can also be factorized into motion and shape matrices, albeit with different ranks. This implies that the temporal factorization of Eq. (3) holds for the matrix W_h as well. In the matrix $W_h = [X, Y]$ there are two columns corresponding to each point and a single row corresponding to each frame, which makes it simpler to use in temporal factorization. Thus, we will henceforth use the matrix $W = \left[\begin{smallmatrix} X \\ Y \end{smallmatrix}\right]$ for spatial multi-body factorization and the matrix $W_h = [X, Y]$ for temporal factorization.

Our approach to subspace-based temporal clustering/factorization can therefore be summarized as follows:

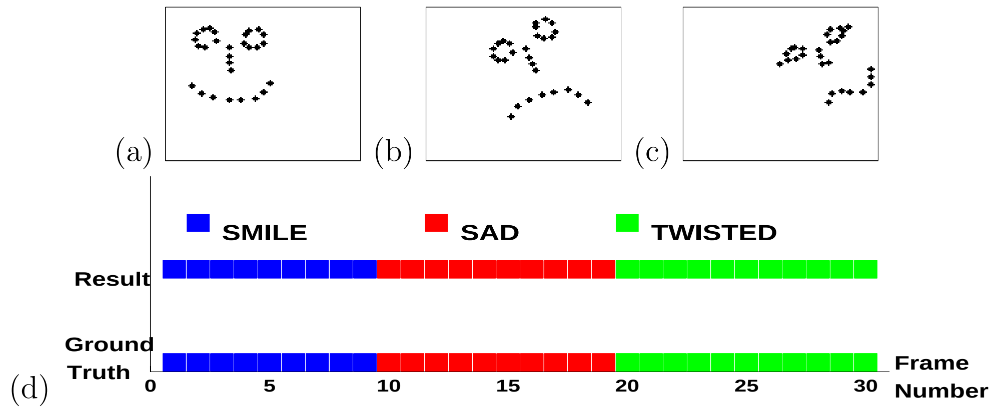


Figure 1. Classifying Synthetic Facial Expressions. *Temporal factorization applied to a 30 frame long synthetic sequence showing a smiley face in three different expressions, while translating and rotating rigidly. (a) Shows frame 1 in which the smiley face is smiling, (b) shows frame 15 in which it is sad and (c) shows frame 30 which displays a "twisted" expression. (d) Temporal factorization result (on the rows of W_h). Setting the number of clusters to 3 resulted in grouping all frames with a smile expression into one cluster (marked in blue on the time bar), all frames with a sad expression into a second cluster (marked in red on the time bar), and all frames with a twisted expression into a third cluster (marked in green on the time bar). Ground truth values are shown for comparison.*

Given a video clip of a dynamic scene:

1. Track reliable feature points along the entire sequence.
2. Place each trajectory into two column vectors (one for the horizontal and one for the vertical component) and construct the correspondence matrix $W_h = [X, Y]$ (see Eq. (1)).
3. Apply any of the existing algorithms for column clustering (e.g., "multi-body factorization" of [6, 7, 9]), but to the matrix W_h^T (instead of W).

3 Physical Meaning of Temporal Factorization

In the following we analyze the properties and characteristics of temporal factorization. For the sake of simplicity we start in Section 3.1 by analyzing the case of a single object. This analysis is extended to the multiple object scenario in Section 3.2.

3.1 The Single Object Case

The temporal factorization scheme suggested above will classify video frames into groups corresponding to independent shapes. When we say "independent shapes" we refer to independence between *rows* of different shape matrices (and *not* between columns/points). Independence between rows of two shape matrices occurs when at least part of the columns in those matrices are different. Recall, that the matrix S corresponding to a rigid set of points is a $4 \times N$ matrix where each column holds the homogeneous coordinates $[X, Y, Z, 1]^T$ of a 3D point. Rigid shape changes can be viewed as the same set of points undergoing a rigid motion, and therefore still have the same shape. On the other hand, *non-rigid* shape changes imply that some of the points move differently than others, i.e., some of the columns of the shape matrix change differently than others. This will lead to a new shape matrix, which is linearly independent of the previous one, and thus to assigning these frames to separate temporal clusters. Note, that every $4 \times N$ shape matrix has a row of 1's there is always partial linear dependence between shape matrices. To overcome that, we can use the Tomasi-Kanade [17] approach for removing the translational component by centering the centroid of the tracked points. Then the row of 1's is eliminated from the shape matrix, and we obtain full linear independence. Alternatively, some of the previously suggested approaches for sub-space segmentation can handle partial dependencies [19]. In particular, we used the spectral clustering approach suggested in [13].

To illustrate the characteristics of temporal factorization, we present in Fig. 1 results of our algorithm applied to a synthetic sequence showing a smiley face in three different expressions "SMILE", "SAD" and "TWISTED", while rotating and translating. As long as the face is smiling, i.e., it's shape is not changing, the rows in the matrix W_h will correspond to the same shape $\tilde{S}_{SMILE}$. However, the change in expression into a sad or a twisted one implies a different shape of the object which cannot be represented as a rigid motion change. Instead we will obtain new shape matrices $\tilde{S}_{SAD}$ and $\tilde{S}_{TWISTED}$ so that: $W_h = \begin{bmatrix} \tilde{M}_{SMILE} & 0 & 0 \\ 0 & \tilde{M}_{SAD} & 0 \\ 0 & 0 & \tilde{M}_{TWISTED} \end{bmatrix} \begin{bmatrix} \tilde{S}_{SMILE} \\ \tilde{S}_{SAD} \\ \tilde{S}_{TWISTED} \end{bmatrix}$. Clustering the rows of W_h grouped all the "SMILE" frames, the "SAD" frames and the "TWISTED" frames, into separate clusters (see Figure 1.d), being invariant to the rigid motion of the smiley face.

Fig. 2 shows this on a real video sequence. It further illustrates the difference between spatial segmentation/grouping of points based on *motion (column)* clustering, and temporal segmentation/grouping of frames based on *shape (row)* clustering. The sequence shows a hand opening and closing the fingers repeatedly. Feature points on the moving fingers were tracked along the sequence using the KLT tracker [10, 1] and used to construct the correspondence matrix W_h (this was used in all our experiments which required tracking). Factoring the rows of W_h (i.e., the columns of W_h^T) into two clusters resulted in temporal shape-based segmentation of frames: It grouped together all the frames with fingers stretched open into one cluster, and all the frames with fingers folded into a second cluster (see Figs. 2.a,b,c). In contrast, applying the segmentation to the columns of W resulted in spatial motion-based segmentation of points into independently moving objects: It grouped into one cluster the points on the fingers which moved mostly horizontally, and grouped into a second cluster points on the thumb which moved mostly vertically, (see Fig. 2.d). The palm of the hand was stationary and hence was ignored.

3.2 The Multiple Objects Case

So far our analysis focused on the single object case. Note, however, that to obtain temporal factorization our only assumption was that the tracked points can lie in a finite number of configurations/shapes and the transformation between these shapes is non-rigid. Thus, when the scene contains multiple objects, we can still obtain temporal factorization if this assumption holds. This is done by viewing the whole scene as a single complex object. The temporal factorization of Equation (3) still holds but the motion and shape matrices, corresponding to the q'th subset of frames will take a more complex form: $\tilde{W}_q = \tilde{M}_q \tilde{S}_q$ where $\tilde{M}_q = \left[\tilde{M}_{q1}, \ldots, \tilde{M}_{qK}\right]$, $\tilde{S}_q = \begin{bmatrix} \tilde{S}_{q1} & & 0 \\ & \ddots & \\ 0 & & \tilde{S}_{qK} \end{bmatrix}$ and K is the number of objects. Temporal cuts will be detected at non-rigid shape changes of

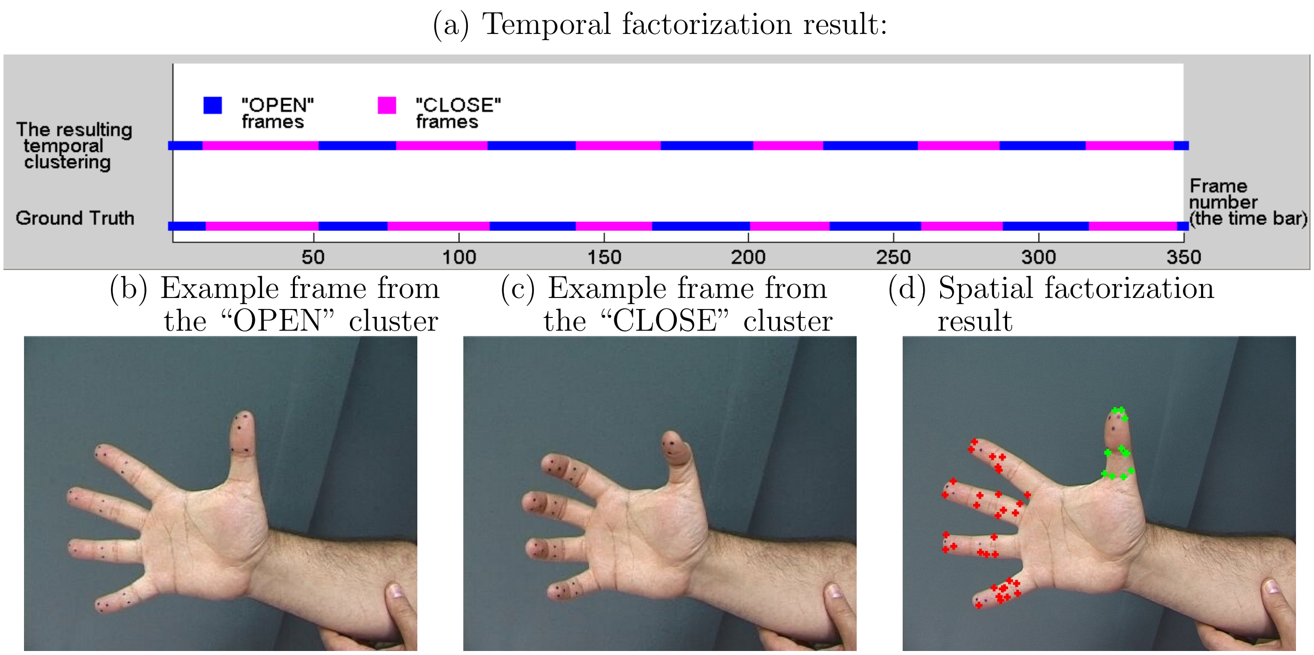


Figure 2. Temporal vs. spatial clustering. *(a) Results of temporal factorization (on the rows of W_h) applied to a sequence showing a hand closing and opening the fingers repeatedly. Setting the number of clusters to 2 resulted in grouping all frames with fingers open into one cluster (marked in blue on the time bar) and all frames with fingers folded into a second cluster (marked in magenta on the time bar). Ground truth values, obtained manually, are shown for comparison. (b),(c) Example frames of the two temporal clusters. (d) Result of spatial factorization (on the columns of W) applied to the same sequence and the same tracked points. This grouped together all the points on the fingers (marked in red), which move mostly horizontally, and classified into a second cluster points on the thumb (marked in green) which move mostly vertically. Note, that since only sparse feature points were tracked along the sequence, the resulting spatial segmentation is highly sparse, whereas the resulting temporal factorization is dense (i.e., all the frames in the video sequence are classified) even though only a sparse set of points is used. Video can be found at http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html*

either of the objects, or at non-rigid changes in the scene organization.

This is illustrated in Fig. 3. The video clip was taken from the movie "Lord of the Rings - Fellowship of the Ring", and shows two hobbits first relaxed and then screaming, while moving their heads. In this case, even-though the scene includes two objects, their non-rigid shape changes are synchronized (i.e., they both change their expression and start screaming at the same time). Note, however, that the rigid head motions of the two hobbits are different and *not* synchronized. The hobbit on the right rotated his head whereas the hobbit on the left did not (see, for example,

Figure 3. Multiple Objects. *Results of temporal factorization (into 2 clusters) applied to a video clip taken from the movie "Lord of the Rings - Fellowship of the Ring". The clip shows two hobbits first calm and then screaming. The shape-based temporal factorization detected the cut between the two expressions and grouped together all the "calm" frames (some example frames are shown in column (a)) separately from all the "scream" frames (some example frames are shown in column (b)). Video can be found at http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html*

Fig. 3.a top and bottom figures). In addition to that, the camera was moving and zooming independently. Nevertheless grouping the rows of W_h into two clusters detected the cut between the two expressions and grouped together all the "calm" frames separately from the "screaming" frames. This is since temporal factorization recognizes non-rigid motions while being invariant to rigid motions.

An additional example of temporal factorization in the multiple-objects case is presented in Figure 4. The video sequence shows a woman and a man exercising independently of each other.

The woman bends left and right whereas the man rocks in a forward-backward motion. The woman's motion is non-rigid and can be viewed as a traversal between 3 different poses (i.e., configurations/shapes): "up", "left" and "right". The man's motion, on the other hand, can be represented as a rigid motion (actually, in the video the man moved his arm non-rigidly but the tracking provided points only on his body thus the captured motion was rigid). Applying our temporal factorization algorithm to points tracked along the sequence resulted, as expected, in a segmentation according to the woman's poses.

Another scenario in which the above assumption holds is when the scene contains multiple rigid objects moving together with the same motion (i.e., rigidly with respect to each other) but every once in a while only one object changes its position with respect to the other objects. In this case the temporal factorization will detect the non-rigid changes in the scene, although the individual objects do not change their shapes and each moves with a rigid motion. An example of such a case is presented in Figure 5.

4 Shape Based Classification Across Sequences

As was shown in the previous sections, the suggested temporal factorization scheme classifies video frames according to shape. So far we have used that for temporal segmentation and clustering of frames within a single video sequence. In this section we show how this approach can be extended to shape-based classification across sequences. In particular we are interested in pose and expression recognition.

The temporal factorization scheme relies on pre-computation of the correspondence matrix W_h. In this matrix each column corresponds to a single point and each row corresponds to a single frame. Given multiple sequences we can stack vertically their corresponding matrices W_h, as if all the frames of all the sequences were taken consecutively by the same camera. The temporal factorization of Eq. (3) holds for this combined matrix as long as the points tracked in all the

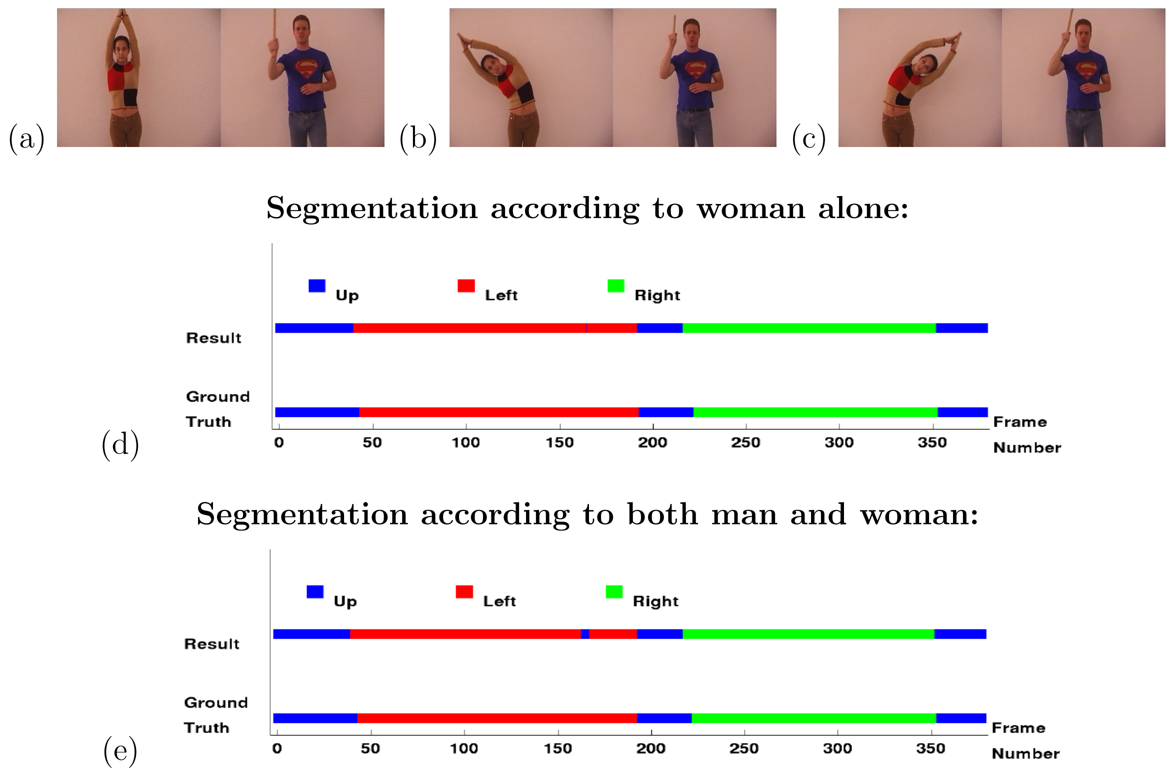


Figure 4. Multiple objects. *Temporal factorization applied to a 376 frame long sequence showing two people exercising. The woman on the left moves non-rigidly: she either stands up-right as shown in (a), bends left as shown in (b) or bends to the right as shown in (c). The man on the right moves almost rigidly in a forward-backward motion. (d) Result of temporal factorization applied only to points tracked on the woman. This grouped frames according to the woman's three poses. (e) Applying the same algorithm to points tracked on both the man and the woman provided almost the same result as that obtained when using only the woman's data. This is since she is the only one performing non-rigid motions.*

sequences are the same points and their corresponding matrix columns are ordered in the same order. When the sequences display different people, this is unlikely to occur. That is, different points will be tracked in different sequences (e.g., the tip of the nose can be tracked in one sequence and not in the other) and their ordering in the corresponding matrix W_h will be arbitrary.

In many cases, however, some of the tracked points correspond to the same features (e.g., the eyes) and thus can be matched as "the same points" across sequences. To select and match only those points which are mutual to all the sequences, we add a preliminary semi-automatic matching step, illustrated in Figure 6. Note, that a single frame from each sequence is sufficient to match the

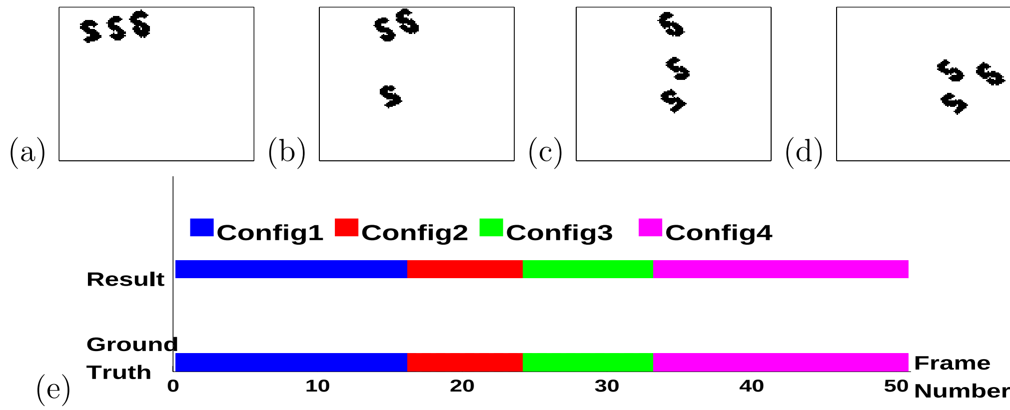
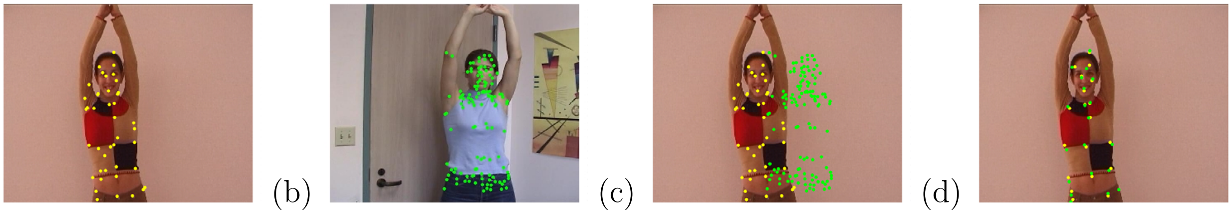


Figure 5. Multiple rigid objects. *Temporal factorization applied to a 50 frame long synthetic sequence showing three rigid objects. The objects move with the same translation and rotation, but appear in 4 different configurations, shown in (a)-(d), which correspond to frames 10, 20, 30 and 40 of the sequence. (e) Applying our temporal factorization algorithm detected the non-rigid changes in the scene and resulted in grouping of frames according to configurations.*

points across sequences. The matching process thus proceeds as follows. First points are tracked independently in each sequence. We select a pose appearing in all the sequences and pick a single representative frame from each sequence corresponding to the selected pose (see Figs. 6.a,b). We further select four body/face features tracked in all the sequences, and manually mark those in each representative frame. For example, in Fig. 6 we marked the eyes and the hips in Fig. 6.a and Fig. 6.b. The selected points are used to estimate a projective transformation which aligns the representative frames. The computed transformation is applied to all the points, thus aligning the representative frames, and only the "mutual points" (i.e., those whose positions across sequences overlap) are kept (see Fig. 6.d). The aligning transformation is then applied to the complete trajectories of the mutual points, which are then used to construct the matrix W_h.

To summarize, given a set of video sequences of different people performing the same activities/expressions we track points in each of the sequences and select only those which were tracked in all of them. The matrix W_h is then constructed by concatenating the tracking results of all the sequences (e.g., given two sequences each 50 frame long the combined matrix W_h will have 100 rows). Applying the temporal factorization algorithm to the combined matrix provides a



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

Figure 6. Finding mutual points across sequences. *(a) The representative frame from one sequence with tracked points marked in yellow. (b) The representative frame from a second sequence with tracked points marked in green. (c) Overlay of the tracked points of both sequences on frame (a). (d) After alignment and keeping only mutual points.*

shape-based classification of frames, rather than a temporal segmentation.

Figure 7 displays results of such pose-based recognition. The input included two sequences, each showing a different woman traversing through a similar set of body poses, but in a different order and with different durations. The shape-based classification described above successfully recognized body poses, independently of the acting person.

Our facial expression recognition experiments were not all that successful. In our experiments we found that some people have such different faces that the classification recognized them as different expressions. Still, in some cases we were able to obtain satisfying results. This is illustrated in Figure 8. Shape based classification was applied to two video sequences displaying two different people in smiling, angry or idle expressions. Setting the number of clusters to 3 provided poor results (see Figure 8.c). On the other hand, applying the same algorithm, to the same set of points, while setting the number of clusters to 4 resulted in high-quality expression based classification (see Figure 8.d). The detected four expressions were “idle”, “smiling”, “angry” of first person and “angry” of the other person. This implies that the angry expressions of the two people were too different to be classified as the same one. Thus, setting the number of clusters to 3 mixed the different expressions, whereas setting the number of clusters to 4 successfully separated between them.

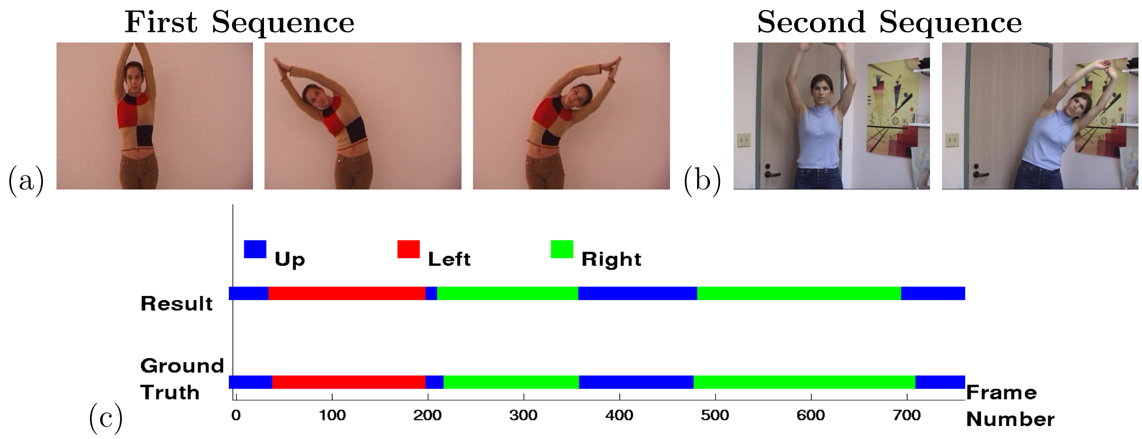


Figure 7. Pose Recognition. *Shape-based clustering applied to two sequences, each showing a different person exercising. The first sequence (clipped off the sequence of Figure 4) shows a woman traversing between three poses: "up", "left" and "right". Corresponding sample frames from the first sequence are shown in (a). The second sequence shows a different woman shifting between the "up" and "right" poses. Corresponding sample frames from the second sequence are shown in (b). (c) Shows the result of shape-based clustering applied to all frames of both sequences simultaneously. The clustering grouped frames correctly according to the three different body poses, although performed by different people. See Section 4 for further details.*

5 Comparing Temporal and Spatial Factorization

In this section we explore the major similarities and differences between the common motion based spatial factorization and our suggested approach to shape based temporal factorization.

Data dimensionality: One of the major difficulties in the multi-body factorization approach is the dimensionality of the data. As was shown by Weiss [20], the method of Costeira & Kanade [6] to multi-body segmentation is equivalent to applying spectral clustering to W^TW, which is an $N \times N$ matrix (N being the number of points). If the number of points is large, then this is a very large matrix. Finding the eigenvectors of such a matrix (which is the heart of spectral clustering) is therefore extremely time consuming. To obtain dense *spatial* factorization of the entire image (e.g., [11]), the number of points N equals the number of pixels in the image which can be extremely large (hundreds of thousands of pixels).

However, this is *not* the case with *temporal* factorization. As explained in Section 2, to obtain

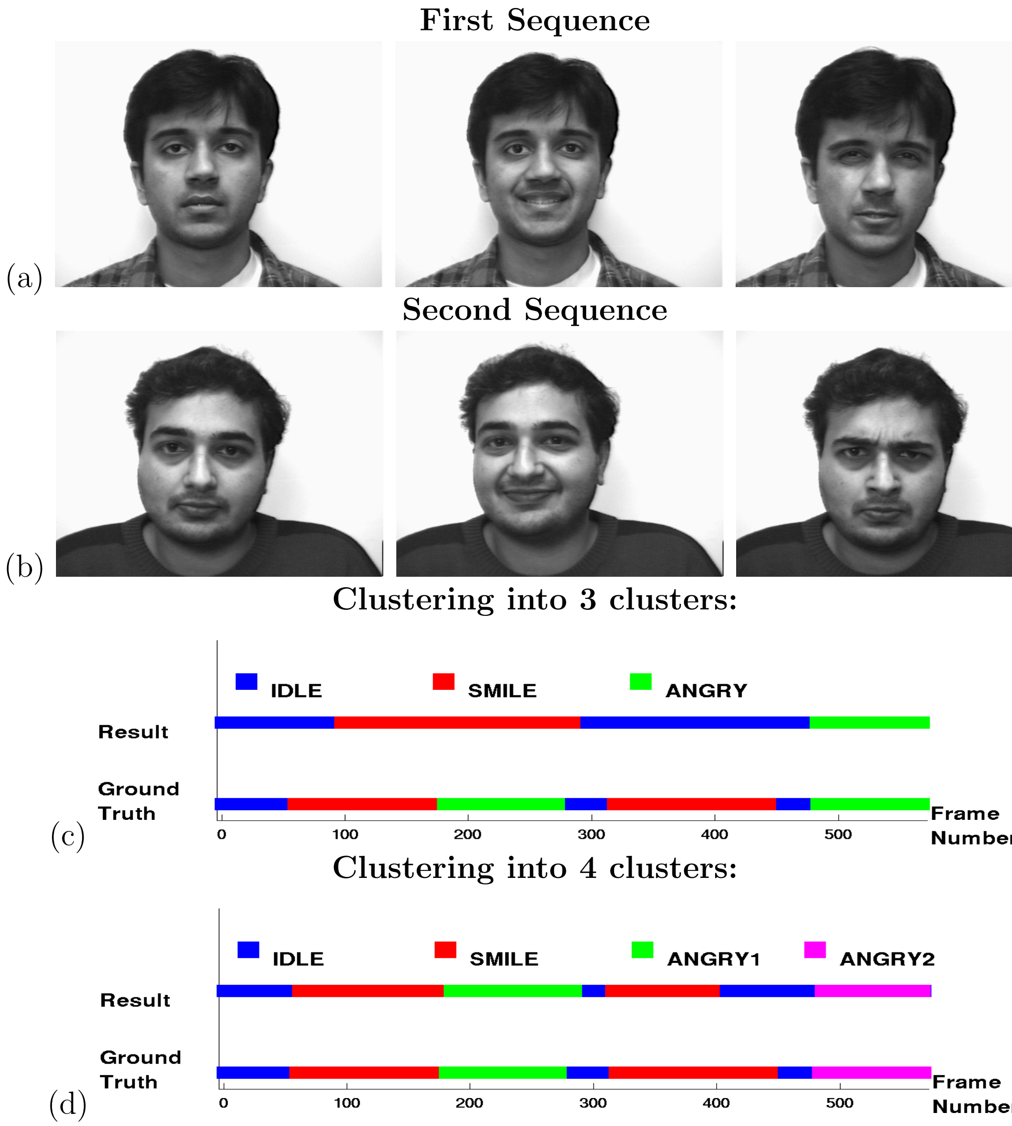


Figure 8. Facial Expression Recognition. *Shape-based clustering applied to two sequences, each showing a different person in three different facial expressions: "idle", "smile" and "angry". Three sample frames from each sequence, corresponding to these three expressions, are displayed in (a) and (b). (c) Shows the result of shape-based clustering applied to all frames of both sequences simultaneously, while setting the number of clusters to 3. A wrong clustering is obtained. (d) Shows high-quality result obtained when setting the number of clusters to 4. This separated between "idle", "smile", "angry" of first person and "angry" of the second person. This explains the bad result when forcing 3 clusters. The "angry" expressions of the two people are too different and thus can not be grouped together, implying that the correct number of clusters to be used here is 4. When setting the number of clusters to 3, the clustering mixed frames corresponding to different expressions. See Section 4 for further details.*

temporal factorization of W_h, we apply the same algorithms suggested for spatial segmentation, but to W_h^T. In other words, this is equivalent to applying spectral clustering [20] to the matrix $W_h W_h^T$ (instead of $W_h^T W_h$). The dimension of $W_h W_h^T$ is $F \times F$, where F is the number of frames in the video clip. Since $F << N$ (F is usually only tens or hundreds of frames) , $W_h W_h^T$ is thus a small matrix, and therefore the temporal factorization is *not* time consuming. Furthermore, while *dense* spatial factorization requires dense flow estimation, dense temporal factorization can be obtained even if only a sparse set of reliable feature points are tracked over time. This is further explained next.

Tracking sparse points vs. dense optical flow estimation: Each column of W contains the trajectory of a single point over time. The data in the matrix W can be obtained either by tracking a sparse set of reliable points or by dense optical flow estimation. Since the spatial "multi-body factorization" clusters the *columns* of W, it will therefore classify only the points which have been tracked. Thus, when only a small set of reliable points is tracked, the resulting spatial segmentation of the image is sparse. Dense spatial segmentation of the image domain requires dense optical flow estimation. This, however, is not the case with temporal segmentation. Since our temporal factorization clusters the *rows* of W_h, there is no need to obtain data for all the points in the sequence. A sparse set of reliable points tracked through all the frames suffices for *dense* temporal factorization. This is because the number of columns in W_h need not be large in order to obtain good row clustering. Results of temporal factorization using a small number of point tracks are shown in Figs. 2 and 3. In Fig. 9 we used dense optical flow measurements to show validity of the approach to both ways of obtaining data. Note, however, that even-though N (the number of points) is large when using optical flow, the computational complexity of the temporal factorization is still low, since the size of $W_h W_h^T$ is independent of N (it depends only on the number of frames F).

Segmentation of $W = \left[\frac{X}{Y}\right]$ Vs. $W_h = [X, Y]$: Recall that $W = \left[\frac{X}{Y}\right]$ and $W_h = [X, Y]$ where the subscript h stands for horizontal stacking of X and Y. The common approaches to multi-body factorization (e.g., [6, 7, 9]) selected carefully tracked feature points, constructed the $W = \left[\frac{X}{Y}\right]$ matrix and clustered its columns. Machline et al. [11] suggested applying multi-body factorization instead to the columns of $W_h = [X, Y]$. This allows to introduce directional uncertainty into the segmentation process, and thus enables dense factorization using unreliable points as well (i.e., dense flow). As noted before, in this matrix, (i.e., W_h) each point has two corresponding columns whereas each frame has a single corresponding row. Thus, when clustering frames (rows) using temporal factorization it is simpler to use the matrix $W_h = [X, Y]$. Note, that when switching from $W = \left[\frac{X}{Y}\right]$ to $W_h = [X, Y]$ the motion matrix completely changes its structure whereas the shape matrix does not. Thus, in spatial multi-body factorization, which is motion based, there is an inherent difference between the two approaches that leads to a different spatial segmentation when using $W_h = [X, Y]$ vs. $W = \left[\frac{X}{Y}\right]$ (see [11]). In contrast, the temporal factorization depends only on shape, thus applying temporal clustering either to $W = \left[\frac{X}{Y}\right]$ or to $W_h = [X, Y]$ will provide the same results. For simplicity we used the $W_h = [X, Y]$ matrix for temporal factorization and the $W = \left[\frac{X}{Y}\right]$ matrix for spatial clustering of points, in all our experiments.

Example: Fig. 9 shows an example of shape vs. motion segmentation using dense optical flow estimation instead of sparse tracking data. The video clip was taken from the movie "Brave Heart", and shows the actor (Mel Gibson) first serious and then smiling while moving his head. The frame-to-frame optical flow was estimated using the robust optical flow estimation software of Michael Black [2] which is described in [4, 3]. The frame-to-frame optical flow fields were composed over time to obtain flow-fields of all frames in the video clip relative to a single reference frame. These flow-fields were then organized in row vectors and stacked to provide the matrix $W_h = [X, Y]$. Applying spectral clustering to the rows of W_h (i.e., applying factorization to the $F \times F$ matrix

$W_h W_h^T$) separated the frames into two clusters: one cluster containing all the "smile" frames, and the other cluster containing all the "serious" frames (see Figs. 9.a,b). For comparison, applying the same clustering algorithm to the columns of W (i.e., applying multi-body factorization to the $N \times N$ matrix $W^T W$) separated between regions with different motions (see Fig. 9.c).

Summary: For further clarification, we summarize in table 1 the observations made in Sections 2 and 5. This provides a summary of the comparison between spatial and temporal factorizations.

	Spatial Factorization	Temporal Factorization
Apply clustering to	$W^T W$	$W W^T$
Data dimensionality	$N \times N$	$F \times F$
Data type	Points (columns)	Frames (rows)
Cluster by	Consistent motions	Consistent shapes
Sparse input	Sparse spatial segmentation	Dense temporal segmentation
Dense input	Dense spatial segmentation	Dense temporal segmentation

Table 1. Comparison summary of spatial factorization vs. temporal factorization

6 A New View on Previous Work

In this section we show that our way of formulating the temporal factorization problem provides a unified framework for analyzing and comparing a number of previously developed independent methods.

The most related work to ours is that of Rui & Anandan [15] who used changes in the frame-to-frame optical flow field to segment activities into their fragments. Rui & Anandan [15] estimated the optical flow field between each pair of consecutive frames and stacked those into a matrix which is highly similar to our $W_h = [X, Y]$ matrix only with displacements instead of positions. They then applied SVD to the matrix, which provided the eigenflows spanning the space of all flow-fields and the coefficients multiplying these basis flow-fields. Temporal cuts were detected at *sign changes of those coefficients*. Their technique can be reformulated in terms of our temporal factorization approach. In our factorization into motion and shape one can view the shape matrix S as being the eigen-vectors spanning the row space and M being the coefficients multiplying

Figure 9. Temporal vs. spatial clustering using dense optical flow. *Results of factorization applied to a sequence taken from the movie "Brave Heart". The actor (Mel Gibson) is serious at first and then smiles while moving his head independently from his expression throughout the sequence. Optical flow was estimated relative to the first frame and the clustering was applied directly to it. We set the number of clusters to 2 for temporal factorization and to 3 for spatial factorization. (a) Sample frames from the first detected temporal cluster, all of which show the actor smiling. (b) Sample frames from the second detected temporal cluster which show the actor serious. (c) Since optical flow was used, we could obtain dense spatial segmentation. This separated between the forehead, the mouth region and a dangling group of hair. These correspond to three independent motions in the sequence: Along the sequence the actor raises his eyebrows and wrinkles his forehead. Independently of that the mouth region deforms when the actor smiles. The group of hair dangles as the head moves, again independently from the other two motions (the motion of the hair at the lower left part of the image can be seen in the frames in (a) and (b)). Video can be found at http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html*

these eigen-vectors. Looking at their work this way shows that they detect cuts at large changes in motion (e.g., shifting from clockwise rotation to counter-clockwise rotation), whereas we detect cuts at non-rigid shape changes and ignore the motion of each shape. Furthermore, reformulating [15] in terms of the temporal factorization approach allows extending it from simple *temporal segmentation* (i.e., detecting cuts) to *temporal clustering*.

Rao & Shah [14] suggested a view-invariant recognition method for complex hand movements. They first obtained hand trajectories (by tracking skin-colored regions) which were sorted according to general structure. Trajectories of similar structure were recognized as the same action by using a low-rank constraint on a matrix constructed from the tracks coordinates. This constraint is equivalent to the one we use for separating between shapes. We detect temporal cuts at increases of the rank and cluster the rows into groups of low rank, i.e., we group frames with the same (or similar) shape.

In a completely different context, Bregler et al. [5] obtained non-rigid object tracking using a factorization/subspace based approach. Their work is *not* related to neither spatial segmentation nor temporal factorization. Nevertheless, we found it appropriate to relate to their work since the shape matrix they used in their decomposition bares similarity to our shape matrix in Eq. (3), which can be misleading. There is a significant difference between their decomposition and ours. They assumed that the shape in each frame is a linear combination of all key-shapes whereas we associate a separate shape with each temporal cluster of frames.

7 Conclusions

This paper explored the properties of temporal factorization of the correspondence matrix W and its duality to spatial factorization of the same matrix. It has been shown that temporal factorization provides a temporal segmentation and clustering of frames according to non-rigid changes in shape, for single object, multiple objects and even across sequences. This approach is unique in the sense that most existing temporal segmentation methods cut the video according to changes in appearance or changes in motion (as opposed to changes in shape).

To obtain such temporal clustering one need *not* develop any new segmentation/clustering scheme but instead can utilize existing algorithms suggested for spatial segmentation. It was further shown that dense spatial segmentation requires dense optical flow estimation whereas a small number of tracked points suffices to obtain a dense temporal clustering of frames, i.e., a classification of *all* the frames in the video clip. Furthermore, the dimensionality of the data, which is one of the major difficulties in spatial multi-body factorization, is significantly smaller for temporal segmentation.

The fact that the same factorization framework can be used for spatial segmentation and for temporal segmentation opens new possibilities that may lead to a combined approach for simultaneous spatio-temporal factorization.

8 Acknowledgments

This work was supported by the European Commission (VIBES project IST-2000-26001).

We wish to thank Yaser Yacoob for the video sequences used for facial expression recognition in Figure 8.

References

[1] S. Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker. http://robotics.stanford.edu/birch/klt/.

[2] M. J. Black. Dense optical flow: robust regularization. http://www.cs.brown.edu/people/black/.

[3] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *International Conference on Computer Vision*, pages 231–236, Berlin, Germany, 1993.

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan. 1996.

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 690–696, 2000.

[6] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *International Conference on Computer Vision*, pages 1071–1076, Cambridge, MA, June 1995.

[7] C.W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 2(29):133–150, 1998.

[8] M. Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, July 2002.

[9] K. Kanatani. Motion segmentation by subspace separation and model selection. In *International Conference on Computer Vision*, volume 1, pages 301–306, Vancouver, Canada, 2001.

[10] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.

[11] M. Machline, L. Zelnik-Manor, and M. Irani. Multi-body segmentation: Revisiting motion consistency. In *Workshop on Vision and Modelling of Dynamic Scenes (With ECCV'02)*, Copenhagen, June 2002.

[12] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Visual Databas Systems II, IFIP*, 1992.

[13] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *In Advances in Neural Information Processing Systems 14*, 2001.

[14] C. Rao and M. Shah. Motion segmentation by subspace separation and model selection. In *International Conference on Computer Vision*, volume 1, pages 301–306, Vancouver, Canada, 2001.

[15] Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2000.

[16] S. Swanberg, D.F. Shu, and R. Jain. Knowledge guided parsing in video databases. In *SPIE*, 1993.

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

[18] L. Torresani, D.B. Yang, E.J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 493–500, 2001.

[19] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[20] Y. Weiss. Segmentation using eigenvectors: A unifying view. In *International Conference on Computer Vision*, pages 975–982, Corfu, Greece, September 1999.

[21] L. Zelnik-Manor and M. Irani. Event-based video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001.

[22] H. Zhang, A. Kankanhali, and W. Smoliar. Automatic partitioning of full-motion video. In *Multimedia Systems*, 1993.