Multi-body Factorization With Uncertainty: Revisiting Motion Consistency

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

Moshe Machline and Michal Irani
(`michal.irani@weizmann.ac.il`)
The Weizmann Institute of Science
Rehovot, Israel

Abstract.
   Dynamic analysis of video sequences often relies on the segmentation of the sequence into regions of consistent motions. Approaching this problem requires a definition of which motions are regarded as consistent. Common approaches to motion segmentation usually group together points or image regions that have the *same* motion between successive frames (where the same motion can be 2D, 3D, or non-rigid). In this paper we define a new type of motion consistency, which is based on temporal consistency of behaviors across multiple frames in the video sequence. Our definition of consistent "temporal behavior" is expressed in terms of multi-frame linear subspace constraints. This definition applies to 2D, 3D, and some non-rigid motions without requiring prior model selection. We further show that our definition of motion consistency extends to data with directional uncertainty, thus leading to a dense segmentation of the entire image. Such segmentation is obtained by applying the new motion consistency constraints directly to covariance-weighted image brightness measurements. This is done without requiring prior correspondence estimation nor feature tracking.

1. Introduction

Common approaches to motion-based segmentation usually track *sparse* feature points and group together those which have the same 3D motion, e.g., (Torr, 1998; Torr and Zisserman, 1998; Costeira and Kanade, 1995; Gear, 1998; Ichimura, 2000; Boult and Brown, 1991). Approaches which provide dense segmentation were usually limited to handling 2D motions, e.g.,(Wang and Adelson, 1993; Ayer and Sawhney, 1995; Irani

et al., 1992). In this work we suggest an approach for *dense* multi-body segmentation which provides a unified treatment of 2D motions, 3D motions and some non-rigid motions.

We define a new type of motion consistency which is based on temporal constraints: a set of points are grouped together as a single object when they have a consistent "behavior" over time. Our definition of consistent "temporal behavior" is expressed in terms of the multi-frame linear subspace constraints used in (Irani, 2002) for flow estimation of a single object. These subspace constraints allow grouping together points moving with *different motions*, as long as their motions change over time in the *same pattern*.

Moreover, using subspace constraints allows introducing directional uncertainty information, as was shown in (Anandan and Irani, 2002) for the single object case. Applying these new multi-body subspace constraints directly to covariance-weighted image brightness quantities gives rise to a segmentation of the entire image (every pixel) into multiple objects. This does not require prior correspondence estimation or feature tracking, nor does it require prior model selection (e.g., 2D, 3D, or non-rigid).

1.1. BACKGROUND ON FACTORIZATION METHODS

Let $I_1, \ldots, I_F$ denote a sequence of F frames with N pixels in each frame. Let (u_i^f, v_i^f) denote the displacement of pixel (x_i, y_i) in frame I_f ($i = 1, \ldots, N$, $f = 1, \ldots, F$). Let U and V denote two $F \times N$ matrices constructed from the displacements of all the image points across all frames:

$$
U = \begin{bmatrix} u_1^1 & \cdots & u_N^1 \\ & \vdots & \\ u_1^F & \cdots & u_N^F \end{bmatrix}_{F \times N} \qquad V = \begin{bmatrix} v_1^1 & \cdots & v_N^1 \\ & \vdots & \\ v_1^F & \cdots & v_N^F \end{bmatrix}_{F \times N} \tag{1}
$$

Each row in these matrices corresponds to a single frame, and each column corresponds to a single point.

Stacking the matrices U and V of Eq. (1) vertically results in a $2F \times N$ matrix $\begin{bmatrix} U \\ V \end{bmatrix}$ where each column is associated with the displacements of a single point across all images/frames. Previous work on subspace-based multi-body segmentation/factorization (e.g., (Costeira and Kanade, 1995; Gear, 1998; Ichimura, 2000; Boult and Brown, 1991)) decomposed the space spanned by the columns of $\begin{bmatrix} U \\ V \end{bmatrix}$ into linearly independent sub-spaces. This was done by permuting and grouping

the columns of $\left[\frac{U}{V}\right]$ into sub-matrices of lower ranks, such that the sub-spaces associated with the different sub-matrices are linearly independent. All columns of a single such sub-matrix (sub-space) of $\left[\frac{U}{V}\right]$ correspond to all the points of a single independently moving object. Such a decomposition was often obtained (e.g., (Costeira and Kanade, 1995; Gear, 1998)) by a factorization of the matrix $\left[\frac{U}{V}\right]$ into a product of two matrices: $\left[\frac{U}{V}\right] = M_{2F \times (r_1 + \cdots + r_K)} S_{(r_1 + \cdots + r_K) \times N}$, where M is the matrix of motions of all objects and S is a block diagonal "shape" matrix with blocks of ranks r_k. Each block corresponds to the shape of a single object and r_k is the rank of that object (i.e., the dimensions of its corresponding sub-space). The permutation and grouping of columns of $\left[\frac{U}{V}\right]$ to obtain the desired separation into independently moving objects was obtained by seeking a block-diagonal structure for the matrix S. While the multi-body factorization approaches assumed rigid motions of objects, in (Brand, 2001) and (Bregler et al., 2000) it was shown that the motion of some non-rigid objects also resides in low-dimensional linear sub-spaces.

In all the above mentioned cases, the sub-space constraints were applied to the matrix $\left[\frac{U}{V}\right]$. In (Boult and Brown, 1991) and (Irani, 2002) this matrix is referred to as the "trajectory matrix", because each *column* in this matrix contains the displacements of a single point across all frames in the sequence, i.e., its trajectory. Irani further investigated (Irani, 2002) the meaning of stacking the matrices U and V horizontally. This gives rise to a $F \times 2N$ matrix $[U, V]$, where each *row* contains the flow field (i.e., displacements of all points) between a single pair of frames (typically between a reference frame and one of the other frames), and is therefore referred to here as the "flow-field matrix" (coined the "displacement-field matrix" in (Irani, 2002)).

In (Irani, 2002) and later in (Anandan and Irani, 2002) it was shown that when using the flow-field matrix $[U, V]$, directional uncertainty can be added to the factorization process. This approach was extended by (Torresani et al., 2001) and (Brand, 2001) to non-rigid objects. However, all the above mentioned methods assumed that all columns of the flow-field matrix $[U, V]$ belong to a *single* (rigid or non-rigid) object/scene.

1.2. What's new in this paper?

In this paper we extend the notion of factorization with uncertainty to multiple (rigid or non-rigid) moving objects[1]. In particular, we show that:

1. Applying the subspace constraints (factorization) to the flow-field matrix $[U, V]$ (instead of the trajectory matrix $\left[\frac{U}{V}\right]$) gives rise to a new and interesting definition of motion consistency, which is different from the traditional motion consistency resulting from $\left[\frac{U}{V}\right]$.

2. Applying the multi-body segmentation algorithm of (Costeira and Kanade, 1995; Gear, 1998) to the *flow-field matrix* $[U, V]$ (instead of the trajectory matrix $\left[\frac{U}{V}\right]$) gives rise to multi-body factorization with directional uncertainty.

3. We show how this gives rise to applying multi-body factorization directly to brightness quantities without prior correspondence estimation. This leads to dense segmentation of the entire image region (pixel by pixel).

The next three sections are dedicated to presenting these three contributions: Section 2 explores the meaning of motion consistency which results from factorizing the flow-field matrix $[U, V]$. Section 3 shows how directional uncertainty can be folded into the factorization of $[U, V]$ (as opposed to $\left[\frac{U}{V}\right]$ which does *not* support directional uncertainty). Section 4 explains how the resulting covariance-weighted factorization can be translated into a direct intensity-based multi-body factorization. This gives rise to dense segmentation with no prior correspondence estimation.

2. Revisiting Motion Consistency

As explained in Section 1.1 clustering the columns of the trajectory matrix $\left[\frac{U}{V}\right]$ (as was done by (Costeira and Kanade, 1995; Gear, 1998; Ichimura, 2000; Boult and Brown, 1991)) captures the dependency between *trajectories* of different points. In this section we investigate the meaning of clustering the columns of the *flow-field* matrix $[U, V]$, and compare the two approaches.

Given two objects and the matrices U_1, V_1, U_2, V_2 of their multi-point multi-frame displacements, we examine when these objects will be grouped together as having consistent motions and when they will be separated, using subspace constraints on either $\left[\frac{U}{V}\right]$ or $[U, V]$. We start by providing a general rank-rule for understanding subspace based segmentation. This rank-rule and its accompanying observations apply to both approaches, and serves as the basis to the analysis of the similarities and differences between the two approaches, which will be given next.

Observations on Subspace Based Segmentations: Let W_1, W_2 be either the trajectory sub-matrices corresponding to the two objects $\{W_1 = \left[\frac{U_1}{V_1}\right]$ and $W_2 = \left[\frac{U_2}{V_2}\right]\}$ or the flow-field sub-matrices of the two objects $\{W_1 = [U_1, V_1]$ and $W_2 = [U_2, V_2]\}$. Let π_1, π_2 be the linear subspaces spanned by the columns of W_1 and W_2, respectively, and let $r_1 = rank(W_1)$ and $r_2 = rank(W_2)$. The two subspaces π_1 and π_2 can lie in three different configurations:

Linear Independence: When π_1 and π_2 are two disjoint linear subspaces (i.e., $\pi_1 \cap \pi_2 = \{0\}$), then $rank([W_1, W_2]) = r_1 + r_2$.

Equality and Inclusion: When one subspace is a subset of (or equal to) the other (w.l.o.g., $\pi_2 \subseteq \pi_1$), then $W_2 = W_1 A$, and $rank([W_1, W_2]) = max(r_1, r_2)$.

Partial Linear Dependence: When π_1 and π_2 partially intersect (i.e., $\{0\} \underset{\neq}{\subseteq} \pi_1 \cap \pi_2 \underset{\neq}{\subseteq} \pi_1 \cup \pi_2$), then $max(r_1, r_2) < rank([W_1, W_2]) < r_1 + r_2$.

To separate between objects, Costeira and Kanade (Costeira and Kanade, 1995) have estimated the SVD of $[W_1, W_2]$, i.e., $[W_1, W_2] = A\Sigma B^T$ (where A and B are unitary matrices) and showed that the "shape interaction matrix" $Q = BB^T = \begin{bmatrix} S_1^T \Lambda_1^{-1} S_1 & 0 \\ 0 & S_2^T \Lambda_2^{-1} S_2 \end{bmatrix}$ has a block diagonal structure. The algorithm they suggested (as well as those suggested in (Ichimura, 2000; Boult and Brown, 1991; Kanatani, 2001)) relied on the block diagonal structure of Q. However, by simple algebraic manipulations we can show that $Q = BB^T = [W_1, W_2]^T [W_1, W_2] = \begin{bmatrix} W_1^T W_1 & W_1^T W_2 \\ W_2^T W_1 & W_2^T W_2 \end{bmatrix}$. This implies that Q has a block diagonal structure only when the matrices W_1 and W_2 are linearly independent. When W_1 and W_2 are partially dependent, then the off-diagonal blocks $W_1^T W_2$ and $W_2^T W_1$ are non-zero. Hence, algorithms like (Costeira and Kanade, 1995; Ichimura, 2000; Boult and Brown, 1991; Kanatani, 2001), which rely on the block diagonal structure of Q, will fail to separate between

such objects. Gear (Gear, 1998) suggested an algorithm which relies on the block-diagonal structure of the reduced-row-echelon form of $[W_1|W_2]$. However, here too, the reduced-row-echelon form will have a block-diagonal structure only when W_1 and W_2 are linearly independent. Gear further suggested (Gear, 1998) a modification to his algorithm which can overcome partial dependencies between W_1 and W_2 in some cases.

We thus conjure the following segmentation rank-rule. In both segmentation approaches (i.e., the existing subspace-based segmentation approaches of $\left[\frac{U}{V}\right]$ or our segmentation of $[U,V]$) the two objects are grouped together or alternatively separated into two objects according to the following (implicit) rank rule [2]. Let $r = rank([W_1,W_2])$, then:

<u>Rank-Rule:</u>

case I	case II	case III
$r = r_1 + r_2$	$max(r_1,r_2) < r < r_1 + r_2$	$r = max(r_1,r_2)$
$\Downarrow$	$\Downarrow$	$\Downarrow$
Two separate objects (in all algorithms)	Algorithm dependent	Single object (in all algorithms)

We make the following observations regarding the Rank-Rule:

1. In some cases the rank of the flow-field matrix $W = [U,V]$ may be higher than the rank of the trajectory matrix $\left[\frac{U}{V}\right]$, and lower in other cases (Irani, 2002). However, the decision whether two objects will be grouped together or separated into two objects depends *solely* on the relative value of the ranks of the sub-matrices W_1 and W_2 and the rank of the combined matrix $[W_1,W_2]$. The decision does *not* depend on the absolute values of these ranks.

2. It can be shown (see Appendix A) that, if two objects are grouped together, by case III of the Rank-Rule, in a trajectory based segmentation $\left[\frac{U}{V}\right]$, they will also be grouped together by case III of the Rank-Rule, in a flow-field based segmentation $[U,V]$, but *not* vice versa! This has interesting implications on multi-body segmentation.

3. Subspace constraints apply to various 2D and 3D motion models (Irani, 2002), and some non-rigid motions (Brand, 2001; Bregler et al., 2000). The decision whether to group two objects together or whether to separate them is based only on the ranks. This does *not* require prior selection of a motion model or a scene geometry.
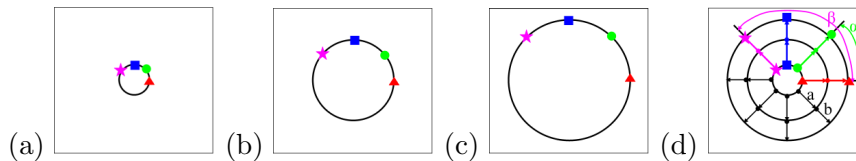
Figure 1. (a)-(c) The 3 frames of a sequence showing a circle enlarging. (d) An overlay image of the three frames and flow vectors for some sample points.

$\left[\frac{U}{V}\right]$ **Segmentation Vs.** $[U, V]$ **Segmentation:** The rank of a matrix is determined by the dimensionality of the linear space spanned by its columns. This is also equal to the dimension of the space spanned by the rows of the matrix (although these spaces are different, their dimensions are the same). To understand the physical meaning of each of the segmentation approaches, we analyze the dependence between columns of the trajectory matrix $\left[\frac{U}{V}\right]$, and the dependence between rows of the flow-field matrix $[U, V]$. The suggested analysis offers an understanding of the information content of each of these matrices and is independent of the specific segmentation algorithm used.

Figure 1 explains the notions of linear dependency of trajectories and linear dependency of flow-fields of a single object. The three frame sequence shows a circle which expands from frame to frame (the radius increases by a between the first and the second frames and by another b in the third frame). It is clear that the trajectory of the top point (marked by a blue square in Fig. 1.d) and the trajectory of the rightmost point (marked by a red triangle in Fig. 1.d) are linearly independent. The top point has a trajectory vector $[a, b, 0, 0]^T$ whereas the rightmost point has a trajectory vector $[0, 0, a, b]^T$. The trajectory of any other point on the circle can be expressed as a linear combination of these two trajectories. For example take the point marked by a green circle in Fig. 1.d: its trajectory vector is: $cos(\alpha)[a, b, 0, 0]^T + sin(\alpha)[0, 0, a, b]^T$. Thus we can write: $\left[\frac{U}{V}\right]_{2F \times N} = \begin{bmatrix} a & 0 \\ b & 0 \\ 0 & a \\ 0 & b \end{bmatrix}_{4 \times 2} C_{2 \times 2}$. The trajectory matrix $\left[\frac{U}{V}\right]$ therefore has rank 2 in this sequence. We next examine the space spanned by the flow-fields of this sequence. Observing the flow-fields in the first pair of frames and in the second pair of frames, it can be seen that they are linearly dependent: $[u_1^2, \ldots, u_N^2 | v_1^2, \ldots, v_N^2] = \frac{b}{a}[u_1^1, \ldots, u_N^1 | v_1^1, \ldots, v_N^1]$. A similar relation will hold for any additional pair of frames in the expanding circle (with different coefficients depending on the rate of

growth). Therefore, $[U|V]_{F \times 2N} = C_{F \times 1}[u_1^1, \ldots, u_N^1 | v_1^1, \ldots, v_N^1]$. This implies that the flow-field matrix $[U|V]$ has rank 1.

In other words, for a single object there exist a set of basis trajectory vectors and a set of basis flow-fields such that the two matrices can be factored into

$$\left[\frac{U}{V}\right] = \underbrace{B}_{\substack{Traj\_Basis \\ (columns)}} \underbrace{C}_{Traj\_Coeff}$$

and

$$[U, V] = \underbrace{C}_{Flow\_Coeff} \underbrace{B}_{\substack{Flow\_basis \\ (rows)}}$$

In the case of multiple objects, as was observed in (Boult and Brown, 1991), segmenting according to the trajectory matrix $\left[\frac{U}{V}\right]$, will group together all the points whose trajectory vectors are spanned by the *same basis* (i.e., $B_1 = B_2$, or when B_1 and B_2 are linearly dependent). Thus, for example, two objects moving with the same 3D motion will be grouped together as one.

Understanding the meaning of grouping points together in the flow-field matrix is less intuitive: Although the subspace constraints are on the *rows* of $[U, V]$, the grouping of points is still done at the level of *columns* (every point has a pair of columns in $[U, V]$). When two objects are grouped together their flow-field vectors at corresponding frames are concatenated into longer flow-field vectors. Two objects will be grouped together into a single object if they have the same coefficients in the linear combination of their individual basis flow-fields (i.e., $C_1 = C_2$, or when C_1 and C_2 are linearly dependent). Note that the two objects can have completely different basis flow-fields, but they will be regarded as one as long as the way in which their flow-field change over time is the same (which is what the coefficients capture). This is what we refer to in this paper as "consistent temporal behavior". For example, this allows grouping together points with *different* motions, as long as the patterns in which their motions change over time is the *same*.

Examples of $\left[\frac{U}{V}\right]$ Segmentation Vs. $[U, V]$ Segmentation: An illustration of the difference between what can be achieved with a multi-point trajectory-based segmentation and a multi-frame flow-field based segmentation can be seen in Figure 2. Fig. 2.a shows a jumping-jack stick figure. All the points on the left arm move together as a rigid body and have linearly dependent trajectories (marked in dark-green

arrows in Fig. 2.b). The points on the right arm also move rigidly (marked in light-green arrows in Fig. 2.b), but in a different motion than that of the left arm. Hence, the trajectory matrix of the combined points of both arms will have a higher rank than any of the trajectory matrices of each individual arm. Therefore, according to the rank rule of Eq. (2), these will *not* be grouped together in $\left[\frac{U}{V}\right]$ segmentation. Similarly, adding the points on the left leg (marked in dark-red arrows in Fig. 2.b) to the trajectory matrix $\left[\frac{U}{V}\right]$ will further increase the rank, and so will adding the points on the right leg (marked in light-red arrows in Fig. 2.b). A segmentation according to the trajectory matrix will therefore segment the four limbs into four separate moving objects. Yet, the flow-field matrix $[U, V]$, which contains all the points on all the limbs, will have the same rank as the flow-field matrix of any single limb. Even though the flow-fields induced by the different limbs are different, they share the same set of coefficients in the frame-to-frame linear dependence of flow-fields (i.e., $C_{right\ arm} = C_{left\ arm} = C_{right\ leg} = C_{left\ leg}$). Therefore, in a segmentation based on the flow-field matrix all the limbs will be grouped as a single object.

Figures 3 and 4 also illustrate the difference between what can be achieved by a trajectory-based segmentation versus a flow-field based segmentation, this time on real sequences. Each of the tested sequences displays a single non-rigid object consisting of a group of rigid sub-parts with different 3D motions. In these examples we tracked points on the sub-parts and analyzed the ranks of the trajectory matrices and of the flow-field matrices both for each separate part as well as for the union of all parts. Fig. 3.e displays the resulting ranks for a video sequence showing a person walking. Points were tracked on three sub-parts: the forearm, the shin (lower-leg) and the head (marked in black in Figs. 3.a-c). The rank of the trajectory matrix for all the points on all parts is higher than the rank of the trajectory matrix for any of the single parts (see Fig. 3.e). On the other hand, the rank of the flow-field matrix for all the points on all parts of this example equals the rank of the flow-fields matrix for the single object with the highest rank (see Fig. 3.e). This implies that a segmentation applied to the flow-field matrix $[U, V]$ will group the forearm, the shin and the head into a single object in this particular sequence.

The resulting ranks for a sequence showing a hand where its five fingers expand and contract simultaneously are displayed in Fig. 4. Points were tracked on each of the fingertips (marked in black in Figs. 4.a-b). The rank of the trajectory matrix for all the tracked points is higher than the rank of the trajectory matrix for a single fingertip (see
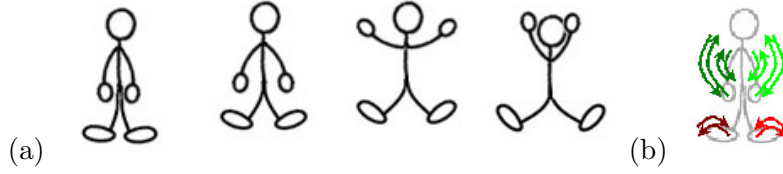
Figure 2. (a) Sample frames from a synthetic sequence showing a stick figure performing jumping-jacks. (b) Example of the trajectories of six different points on the figure, one on the left leg, one on the right leg, two on the left arm and two on the right arm.
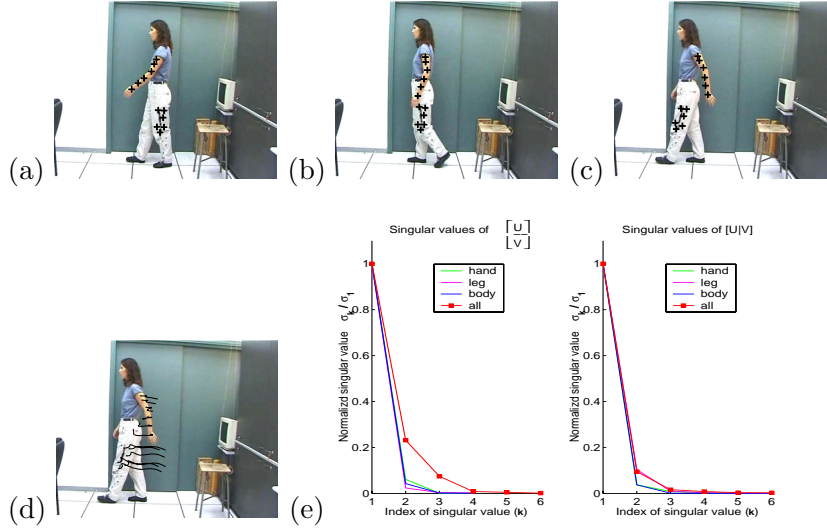

Figure 3. (a)-(c) Sample frames from a sequence showing a person stepping forward. The tracked points are marked in red. (d) The trajectories of the tracked points displayed on a sample image. (e) Graphs showing the singular values of the trajectory matrices and of the flow-field matrices for each object separately and for all points together.

Fig. 4.e). Alternatively, the rank of the flow-field matrix for all the points equals the rank of the flow-field matrix for a single fingertip (see Fig. 4.e). This implies that in this case a segmentation according to the flow-field matrix will group all the fingers into a single object.

3. Multi-Body Factorization with Uncertainty

In the above-discussed segmentation of the trajectory matrix $\left[\frac{U}{V}\right]$, or the flow-field matrix $[U, V]$, we have assumed that feature points have
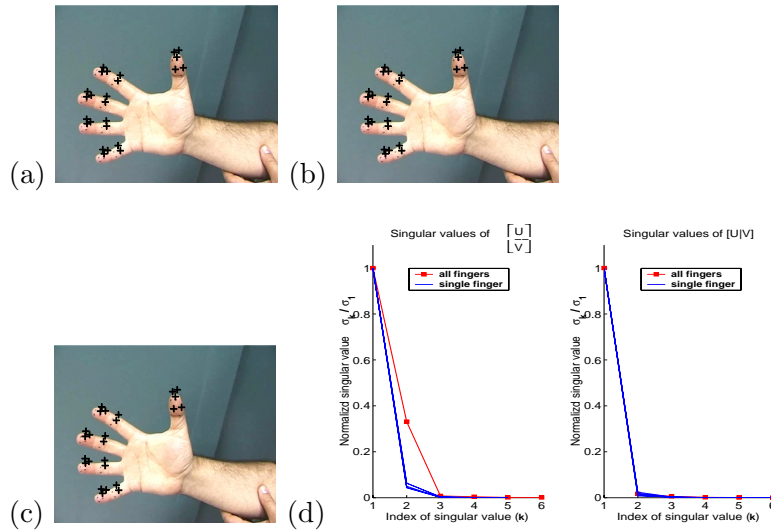
Figure 4. (a)-(b) Sample frames from a sequence of a hand where the fingertips move simultaneously (the hand is squeezing and expanding). The tracked points are marked in black. (c) The trajectories of the tracked points marked on a sample image. (d) Graphs showing the singular values of the trajectory matrices and of the flow-field matrices for each object separately and for all points together.
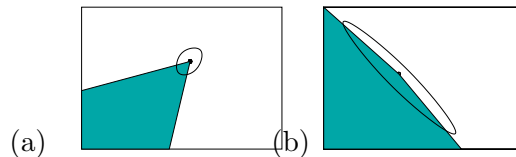

Figure 5. Directional uncertainty indicated by ellipse. (a) Uncertainty of a sharp corner point. The uncertainty in all directions is small, since the underlying intensity structure shows variation in multiple directions. (b) Uncertainty of a point on a flat curve, almost a straight line. Note that the uncertainty in the direction of the line is large, while the uncertainty in the direction perpendicular to the line is small. This is because it is hard to localize the point along the line.

been reliably tracked across all frames. However, in general, only a sparse set of feature points in an image can be assumed to produce such reliable tracking along a sequence. To allow for *dense* segmentation of the entire image (and not only grouping of a sparse set of image points) we introduce the notion of local uncertainty into the segmentation process.

The quality of feature matching depends on the spatial variation of the intensity pattern around each feature, which affects the positional inaccuracy both in the x and in the y components in a correlated fashion. This dependency can be modelled by *directional* uncertainty

(which varies from point to point, as is shown in Fig. 5). Such directional uncertainty can be modelled by a 2×2 inverse covariance matrix (Morris and Kanade, 1998).

(Anandan and Irani, 2002) showed that directional uncertainty can be introduced into the flow-field matrix $[U, V]$, whereas only isotropic uncertainty can be introduced into the trajectory matrix $\left[\frac{U}{V}\right]$. We briefly review the reasons for this next.

Every column in the trajectory matrix $\left[\frac{U}{V}\right]$ corresponds to a different point. To associate a different uncertainty with each point one can multiply the trajectory matrix $\left[\frac{U}{V}\right]$ by a (diagonal) weight matrix A on the right to get: $\left[\frac{U}{V}\right] A$. While this will associate a different uncertainty (weight) to different points (columns), it will associate the same uncertainty (same weight) to the x and y components of a point. This implies that one can introduce only *isotropic* uncertainty information into the trajectory matrix $\left[\frac{U}{V}\right]$. In (Anandan and Irani, 2002) it was shown, however, that multiplying the *flow-field* matrix $[U, V]$ by a weight matrix on the right *does* allow the introduction of directional uncertainty. Multiplying $[U, V]$ by a weight matrix Q on the right (to get $[U, V]Q$) achieves two goals: (i) we can assign different weights to the x and y components of the same point (because they are now in different columns), and, (ii) we can introduce correlation between the x and y components of the same point. The large matrix Q is generated from the values in all the individual 2×2 local inverse covariance matrices (we explain the construction of the matrix Q in the description of the algorithm in Section 4). Hence, using the flow-field matrix $[U, V]$ allows us to introduce *directional uncertainty*. Using the inverse covariance matrices associated with each point allows us to transform the input displacement vectors from the raw data space into a new data space (the "covariance-weighted space"), where the noise is uncorrelated and identically distributed. In the new covariance-weighted data space, corner points and points on lines all have the same reliability, and their new positional components are uncorrelated (this is in contrast with the original data space, where corner points and points on lines had different reliability, and their x and y components were correlated). This allows performing factorization even to pure normal-flow data. For more details see (Anandan and Irani, 2002).

(Anandan and Irani, 2002) dealt with a single object factorization. We would like to extend this reasoning to multi-body segmentation. The quality of the multi-body segmentation depends directly

on the accuracy of the estimated displacements (u_i^f, v_i^f) $(i = 1, \ldots, N$, $f = 1, \ldots, F)$ of the tracked feature points. All previous work on subspace-based multi-body segmentation/factorization (e.g., (Costeira and Kanade, 1995; Gear, 1998; Ichimura, 2000; Boult and Brown, 1991)) relied on carefully tracked *sparse* feature points. In order to get *dense* segmentation we introduce directional uncertainty into the multi-body factorization process.

As explained above, directional uncertainty can only be introduced in the flow-field matrix $[U, V]$, and not in the trajectory matrix $\left[\frac{U}{V}\right]$. Therefore, we first show (Section 3.1) that the multi-body factorization framework of (Gear, 1998) and (Costeira and Kanade, 1995), which was originally developed for $\left[\frac{U}{V}\right]$, applies to $[U, V]$. We then show (Section 3.2) that by introducing directional uncertainty information we can translate these constraints from flow-fields to covariance weighted quantities (i.e., we show that we can get factorization with uncertainty also in the multi-body case).

3.1. Multi-Body Constraints on the Flow-Field Matrix $[U, V]$

The flow-field matrix of a single object can be factored into a multiplication of two matrices: $[U, V] = [M_U, M_V] \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix}$, where M_U, M_V contain motion information and S contains shape information (Anandan and Irani, 2002; Irani, 2002). When the scene contains multiple moving objects, their columns are usually not sorted and are mixed in the flow-field matrix $[U, V]$. Let $[\tilde{U}, \tilde{V}]$ be a matrix obtained by sorting the columns of $[U, V]$ into independent objects. Then it should have the following form:

$$
\left[\tilde{U}, \tilde{V}\right] = [U_1, V_1, \cdots, U_K, V_K]
$$

$$
= \overbrace{[M_{U_1}, M_{V_1}, \cdots, M_{U_K}, M_{V_K}]}^{M} \overbrace{\begin{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_1 \end{bmatrix} & 0 & \cdots & & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & & \cdots & 0 & \begin{bmatrix} S_K & 0 \\ 0 & S_K \end{bmatrix} \end{bmatrix}}^{S} \quad (2)
$$

where K is the number of objects. Note that the combined shape matrix S of the sorted matrix $[\tilde{U}, \tilde{V}]$ is block diagonal, where each block corresponds to a single object.

Thus, when only the *unsorted* flow-field matrix $[U, V]$ is available, its columns can be sorted and grouped into objects by seeking a block diagonal structure of the total "shape" matrix S of Eq. (2). The sorting can be done in the same way suggested by (Gear, 1998) or (Costeira and Kanade, 1995) for the $\left[\frac{U}{V}\right]$ matrix. However, as was shown in Section 2, in the case of the $[U, V]$ matrix, such grouping will lead to grouping points which have *temporally consistent behaviors* (and not necessarily motion). We next show how this can be applied directly to covariance-weighted measurements, thus allowing for segmentation of very inaccurate flow-fields.

3.2. Multi-Body Constraints on Covariance-Weighted Measurements

Let $\begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$ be the 2×2 inverse covariance matrix containing the local uncertainty information at a pixel i ($i = 1, \ldots, N$). Usually, the inverse covariance matrix is symmetric, i.e., $b_i = c_i$. This information can be computed from the underlying local brightness pattern at each pixel in the image (see (Anandan and Irani, 2002)). When all the flow vectors across the entire image sequence are estimated relative to a single reference image frame then, $\begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$ is the same for pixel i across all frames (i.e., when $[u_i^f v_i^f]$ is the flow-vector of pixel i between the reference-frame and frame f). It was shown by (Anandan and Irani, 2002) that one can construct a $2N \times 2N$ matrix $Q = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ where A, B, C and D are diagonal $N \times N$ matrices constructed from the individual coefficients a_i, b_i, c_i and d_i of each pixel. Multiplying the flow-field matrix $[U, V]$ on the right by the matrix Q gives the covariance-weighted measurement matrix $[G, H]$:

$$[G, H]_{F \times 2N} = [U, V]_{F \times 2N} Q_{2N \times 2N}. \tag{3}$$

When the scene contains multiple moving objects, sorting the columns of $[G, H]$ according to objects gives $[\tilde{G}, \tilde{H}]$ such that:

$$[\tilde{G}, \tilde{H}] = [G_1, H_1, \cdots, G_K, H_K] = [U_1, V_1, \cdots, U_K, V_K] \begin{bmatrix} Q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Q_K \end{bmatrix}$$
(4)

where $Q_k = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix}$ are $2N_k \times 2N_k$ matrices and N_k is the number of image points belonging to object k. Note, that the right-hand-side matrix of Eq. (4) has a block-diagonal structure.

Combining Eq. (2) and (4) gives:

$$[\tilde{G}, \tilde{H}] = \overbrace{[M_{U_1}, M_{V_1}, \cdots, M_{U_K}, M_{V_K}]}^{\tilde{M}} \overbrace{\begin{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_1 \end{bmatrix} Q_1 & \cdots & & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & \cdots & \begin{bmatrix} S_K & 0 \\ 0 & S_K \end{bmatrix} Q_K \end{bmatrix}}^{\tilde{L}}$$
(5)

Eq. (5) implies that the sorted matrix $[\tilde{G}, \tilde{H}]$ can be factored into a product of two matrices $[\tilde{G}, \tilde{H}] = \tilde{M}\tilde{L}$, where $\tilde{L}$ is block diagonal. The number of blocks in this representation corresponds to the number of moving objects in the scene, and the rank of each block r_k characterizes the rank of each object. In other words, multi-body factorization can be applied directly to the covariance-weighted flow $[G, H]$, and not only to accurate flow $[U, V]$. In particular, it can also handle pure normal-flow information (which has infinite uncertainty in one direction).

The unsorted covariance-weighted flow-field matrix $[G, H]$ can be constructed from dense optical flow information (which is available for *all* the points) weighted by their local uncertainty (which is also available for all image points). We no longer need to detect and track sparse feature points. Having the $[G, H]$ matrix, its columns can be sorted and grouped into objects (with temporally consistent behaviors) by seeking a block diagonal structure of the right-hand-side matrix of Eq. (5). This can be done using similar methods to those used in (Gear, 1998) or (Costeira and Kanade, 1995).

4. Direct Intensity-Based Multi-Body Segmentation

The covariance-weighted flow matrix $[G, H]$ can be computed in two steps, by first computing the (unreliable) flow matrix $[U, V]$, and then multiplying it by the local directional uncertainty. Alternatively, $[G, H]$ can be computed *directly* from image brightness quantities with no prior flow estimation. We next present an algorithm for multi-body segmentation of video clips by applying sub-space constraints of the flow-field matrix $[U, V]$ *directly* to image brightness measurements. The segmentation is embedded in an iterative coarse-to-fine framework and extends the work of (Irani, 2002) into multiple moving objects. The algorithm that is outlined below does not require prior tracking of points nor any prior optical-flow estimation, and segments the entire image (pixel by pixel) into groups characterized by consistent behavior over time.

In (Lucas and Kanade, 1981) the estimation of the local flow vector $[u_i^f v_i^f]$ of a pixel i at frame f was done by solving the equation:

$$[u_i^f v_i^f] \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}_i = [-\sum I_x I_t^f \quad \sum I_y I_t^f]_i^f \qquad (6)$$

where $\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$ and $[-\sum I_x I_t^f \quad \sum I_y I_t^f]$ are measurable image quantities, in which I_x and I_y are the spatial image derivatives (computed in the reference image) and I_t^f is the temporal image derivative (between frame f and the reference frame). The summation is performed over a local window around pixel i. Under Gaussian noise assumptions the spatial derivative matrix $\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}_i$ can be shown to be the posterior inverse-covariance matrix of the flow-vector $[u_i^f v_i^f]$. Thus, the measurable image quantities $[-\sum I_x I_t^f \quad \sum I_y I_t^f]_i^f$ approximate the covariance-weighted flow information.

Instead of first computing flow and then weighting it by its local directional uncertainty we estimate the covariance-weighted flow $[G, H]$ and the local uncertainty Q (the use of Q in the direct algorithm is explained in the algorithm description in Section 4.1) for *all* pixels directly from image brightness measurements:

$$[g_i^f h_i^f] = [-\sum I_x I_t \quad -\sum I_y I_t]$$

$$\begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

We then construct the unsorted brightness-measurement matrix $[G, H]$.

The above estimation assumes brightness constancy and thus will not perform well under effects of shadows, changing illumination etc. When the changes in brightness are not too rapid we overcome this problem by taking into account in the segmentation only a small number of frames in which the brightness constancy assumption holds. We then use a sliding temporal window to obtain a segmentation for all frames.

Eq. (5) implies that segmenting the entire image (all pixels) into independent objects can be done by sorting the columns of the brightness-measurement matrix $[G, H]$. We obtain such a sorting by finding the reduced row-echelon form of $[G, H]$ using a method similar to that suggested in (Gear, 1998)[3]. However, since $[G, H]$ is a large $F \times 2N$ matrix, which in practice has a much lower rank r, we can first factor $[G, H]$ into two rank-r matrices using SVD: $[G, H] = M_{F \times r} \cdot L_{r \times 2N}$, and then find the reduced row-echelon form of the smaller matrix L. When M is full ranked then the matrix L has the same reduced row-echelon form as $[G, H]$ (see (Gear, 1998) for more details).

Interestingly, for the purpose of segmentation, accurate knowledge of the rank r and the block ranks $r_1, \ldots, r_K$ is *not* necessary. In fact, even if the rank of the matrices M and L (and hence also the rank of $[\hat{G}, \hat{H}] = ML$) is lower than the true rank of $[G, H]$, correct segmentation is still possible. This is because clustering (segmentation) is a "competitive" process between the different objects which is often resolved already by a few dominant basis vectors, and may not require the use of the entire basis. This is unique to the segmentation task, and is not true for shape and motion recovery tasks, which require accurate knowledge of the true ranks $r_1, \ldots, r_K$.

The measured image quantities $a_i, b_i, c_i, d_i, g_i^f, h_i^f$ of Eq. (6) are obtained from the linearized brightness constancy equation. However, this linearization is a good approximation only for small displacements (u_i^f, v_i^f). To handle larger displacements, we apply our segmentation scheme within a multi-frame multi-scale (pyramid) data structure. Large displacements at high resolution levels translate to small displacements at coarse resolution levels. Our segmentation scheme can therefore be applied at coarse resolution levels, but this will only provide coarse segmentation. To refine it, the process must be propagated to higher resolution levels. This is achieved by an iterate-warp coarse-to-fine framework similar to the one used in (Irani, 2002) for a single rigid scene. Since warping requires intermediate flow estimation, more accurate knowledge of the ranks $r_1, \ldots, r_K$ is required. These ranks are automatically detected from the brightness matrices $[G_k, H_k]$ for each

of the segmented objects extracted in the previous iteration. Note that this does *not* require prior model selection (2D, 3D or non-rigid), as no 3D shape or motion information is recovered (see (Irani, 2002) for more details). We next summarize the algorithm.

4.1. SUMMARY OF THE DIRECT MULTI-BODY SEGMENTATION ALGORITHM:

(1) Construct a Gaussian pyramid for all image frames.
(2) For each iteration at each pyramid level (starting at the lowest resolution level) do:

- Compute matrices G, H and Q of Eq. (3) directly from brightness quantities.

- Factorize $[G, H]_{F \times 2N}$ into $M_{F \times r} \cdot L_{r \times 2N}$, thus reducing dimensionality and noise.

- Find the reduced row echelon form of L.

- Sort the reduced row echelon form of L into groups of columns which correspond to the same object. This defines the correct sorting of $[G, H]$.

- Project $[G_k, H_k]$ $(k = 1, \ldots, K)$ of each object onto a low-dimensional linear subspace of dimension r_k to reduce noise and get $[\hat{G}_k, \hat{H}_k]$.

- Estimate the displacements from the equation: $[U_k, V_k] \cdot Q_k = [\hat{G}_k, \hat{H}_k]$ where $k = 1 \ldots K$ (see (Irani, 2002)).

- Warp all frames towards the reference frame according to the estimated displacements (for all objects).

(3) Iterate step 2 several times (typically 4-5 times) in each resolution and propagate the result to the next (higher) resolution level in the pyramid.

For computational efficiency, the coarse-to-fine process is often halted at an intermediate resolution level, (For example, in our experiments we usually stopped at the pyramid level corresponding to $\frac{1}{4}$ resolution in image width and height, i.e., at $\frac{1}{16}$ of the image size N). The penalty for this is that the *boundaries* of the detected segments (objects) are not recovered accurately.

5. Results

We tested our direct multi-body segmentation algorithm on video sequences with different types of induced camera motions (2D planar / 3D), and different types of object motions (rigid / non-rigid). The sequences consist of 40-80 frames each, and with approximate image size of $\sim 300 \times 350$. As mentioned above, the coarse-to-fine segmentation process was stopped at a pyramid resolution corresponding to $\frac{1}{16}$ of the image size.

The first sequence consists of a globe rolling on a track situated in a 3D scene (Figs. 6.a-d. See sequence *globus* at web-site). The object is undergoing a rotation and translation while the camera is translating from right to left, back and forth, inducing 3D parallax effects of the background scene (e.g., note the parallax between the two tripods: they occlude each other in Fig. 6.b and are both visible in Fig.6.c). Applying our algorithm to the scene produced a segmentation into two separate objects (see Figs. 6.e-g), one which corresponds to the globe and the second which corresponds to the background scene. The background was grouped into a single object even though no explicit 3D model was assumed for the background motion. This is because although pixels with different depths have different flows, the pattern of changes over time in their flows are the same (i.e., they share the same coefficients over time in the factorization of the $[U, V]$ flow-matrix).

The second example consists of an outdoor scene in which a helium balloon drifts up with changing wind directions (Figs. 7.a-c. See sequence *baloon* at web-site). The camera is again translating from left to right and back causing parallax effects (e.g., the tree trunk is occluded by the woman on the bench in Fig. 7.a and is visible in Fig.7.b). The result of applying our algorithm (Fig. 7.d-f) detects the balloon as one object and groups the 3D background scene into a second object.

The third sequence shows a hand with the fingers expanding and contracting making a squeezing motion, and a plant with dense leaves in the background (Figs. 8.a-c. See sequence *hand* at web-site). Each finger moves in an articulated motion while the motion of all fingers is non-rigid. In addition, the camera is rotating around the Z axis. As predicted by the analysis in Section 2, the flow-field based segmentation algorithm grouped together all the fingers into a single object, and all the parts which undergo pure camera motion (including the arm and the palm of the hand, which were stationary relative to the background) into a second object (Figs. 8.d-f).

The forth experiment was done on a sequence showing a large piece of cloth being folded and unfolded in a natural waving motion, while a

branch of leaves is being waved up and down in front of it (Figs. 9.a-c. See sequence *cloth* at web-site). The motion of the cloth is non-rigid. The two sides of the cloth are waving from outside towards the center, while the center top part of the cloth moves vertically (i.e., the top center of the cloth is raised as the cloth is stretched in Fig. 9.a and is lowered as the cloth is folded in Fig. 9.b). However, all the parts of the cloth have *consistent temporal behavior*. Indeed applying the segmentation process of Section 4 segmented the scene into two objects, one which corresponds to the leaves and the second which corresponds to the cloth (see Figs. 9.d-f). The textureless background region (marked in green in Fig. 9.f) was ignored in this example. Note that the different regions of the cloth were grouped together, even though no non-rigid motion model was specified.

The boundaries of the segmentation results in some of the examples that we have used can be considerably improved by proceeding to higher resolution levels of the pyramid, and by utilizing more sophisticated clustering methods than the naive clustering approach which we implemented.

6. Conclusions

In this paper we presented an analysis of motion consistency in subspace based segmentation methods. This analysis led to a new definition for motion consistency which can handle both rigid and non-rigid objects in a single framework. Based on this, we suggested an algorithm which groups regions with consistent behavior along time. The suggested algorithm applies the new motion consistency constraints directly to covariance-weighted image brightness measurements resulting in a dense spatial segmentation. This is done without requiring prior correspondence estimation nor feature tracking and without prior knowledge on the types of motions in the sequence.
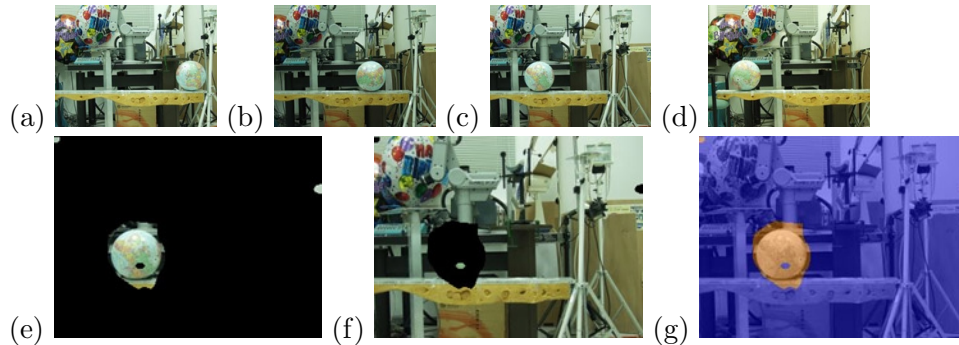
7. Acknowledgments

Figure 6. Segmentation under 3D parallax: *(a)-(d) Four sample frames of a sequence showing a globe rolling on a track while the camera translates (see attached sequence* globus.mpg*). (e)-(g) Results of the direct multi-body segmentation algorithm. (e) Shows the first object, (f) shows the second object and (g) shows an overlay of the segmentation result on the reference image. For sequences and results see* `http://www.wisdom.weizmann.ac.il/∼vision/MultiBodySeg.html`*.*

Figure 7. Segmentation under 3D parallax: *(a)-(c) Three sample frames of a sequence showing a helium balloon drifting with the wind while the camera translates (see attached sequence* baloon.mpg*). (d)-(f) Results of the direct multi-body segmentation algorithm. (d) Shows the first object, (e) shows the second object and (f) shows an overlay of the segmentation result on the reference image. For sequences and results see* `http://www.wisdom.weizmann.ac.il/∼vision/MultiBodySeg.html`*.*
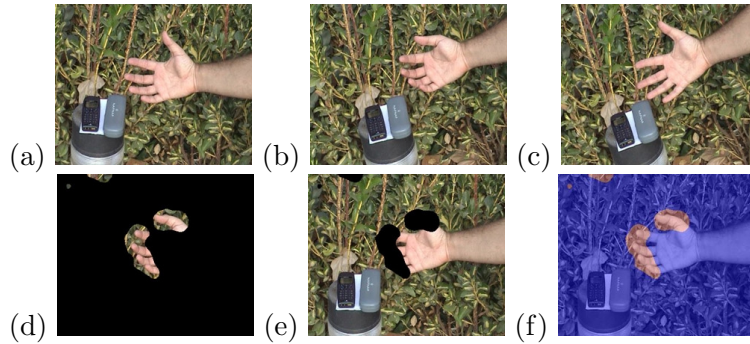
Figure 8. Segmentation under articulated motion: *(a)-(c) Three sample frames of a sequence showing a hand with the fingers expanding and contracting while the camera translates and rotates (see attached sequence* hand.mpg*). (d)-(f) Results of the direct multi-body segmentation algorithm. (d) Shows the first object, (e) shows the second object and (f) shows an overlay of the segmentation result on the reference image. For sequences and results see* http://www.wisdom.weizmann.ac.il/∼vision/MultiBodySeg.html*.*
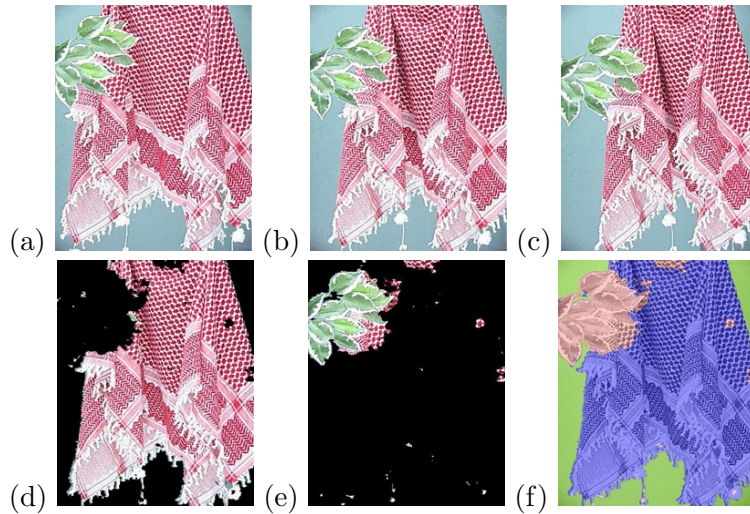

Figure 9. Segmentation under non-rigid motion: *(a)-(c) Three sample frames of a sequence showing a cloth moving non-rigidly (folded and stretched) and a branch waving independently, while the camera remains still (see attached sequence* cloth.mpg*). (d)-(f) Results of the direct multi-body segmentation algorithm. (d) Shows the first object, (e) shows the second object and (f) shows an overlay of the segmentation result on the reference image. For sequences and results see* http://www.wisdom.weizmann.ac.il/∼vision/MultiBodySeg.html*.*

Appendix

A. Relation between $\left[\frac{U}{V}\right]$ and $[U, V]$ Segmentation

In this section we give the proof of observation 2 of Section 2.

Claim 1: *If two objects are grouped together by case III of Eq. (2) in a trajectory based segmentation of $\left[\frac{U}{V}\right]$, they will also be grouped together by case III of Eq. (2) in a flow-field based segmentation of $[U, V]$ (but* not *necessarily vice versa).*

Claim 2: *Equivalently, if two objects are separated by case I of Eq. (2) in a flow-field based segmentation of $[U, V]$, they will also be separated by case I of Eq. (2) in a trajectory based segmentation of $\left[\frac{U}{V}\right]$ (but* not *necessarily vice versa).*

Proof:

First direction:

Without loss of generality, assume $rank(\left[\frac{U_1}{V_1}\right]) \geq rank(\left[\frac{U_2}{V_2}\right])$.

Claim 1: Assuming that the two objects are grouped together according to case III of Eq. (2) in a $\left[\frac{U}{V}\right]$ segmentation, then:

$$rank(\left[\frac{U_1}{V_1}, \frac{U_2}{V_2}\right]) = rank(\left[\frac{U_1}{V_1}\right])$$
$$\Downarrow$$
$$\exists C_{N_1 \times N_2} \text{ such that } \left[\frac{U_2}{V_2}\right] = \left[\frac{U_1}{V_1}\right] C$$
$$\Downarrow$$
$$[U_1, V_1, U_2, V_2] = [U_1, V_1] \begin{bmatrix} I & 0 & C & 0 \\ 0 & I & 0 & C \end{bmatrix}$$
$$\Downarrow$$
$$rank([U_1, V_1, U_2, V_2]) = rank([U_1, V_1])$$
$$\Downarrow$$

> The two objects will also be grouped together according to case III of Eq. (2) in a $[U, V]$ segmentation.

Claim 2: Assuming that the two objects are separated according to case I of Eq. (2) in $[U, V]$ segmentation, then:

$$rank([U_1, V_1, U_2, V_2]) = rank([U_1, V_1]) + rank([U_2, V_2])$$
$$\Downarrow$$
$$columns([U_1, V_1]) \ \text{ lin. ind. } \ columns([U_2, V_2])$$
$$\Downarrow$$
$$columns\left(\left[\frac{U_1}{V_1}\right]\right) \ \text{ lin. ind. } \ columns\left(\left[\frac{U_2}{V_2}\right]\right)$$
$$\Downarrow$$
$$rank\left(\left[\frac{U_1}{V_1}, \frac{U_2}{V_2}\right]\right) = rank\left(\left[\frac{U_1}{V_1}\right]\right) + rank\left(\left[\frac{U_2}{V_2}\right]\right)$$
$$\Downarrow$$

> The two objects will also be separated according to case I of Eq. (2) in a $\left[\frac{U}{V}\right]$ segmentation.

Second Direction: Here we will show that the other direction is not true, i.e., we give an example where two objects are grouped together in a flow-field based segmentation but separated in a trajectory based segmentation. Note that this contradicts both the reverse direction of claim 1 and of claim 2.

$$\text{Let} \quad rank([U_1, U_2]) = rank(U_1) + rank(U_2)$$
$$\text{(i.e., } U_1 \text{ and } U_2 \text{ are linearly independent)}$$
$$\Downarrow$$
$$rank\left(\left[\frac{U_1}{V_1}, \frac{U_2}{V_2}\right]\right) = rank\left(\left[\frac{U_1}{V_1}\right]\right) + rank\left(\left[\frac{U_2}{V_2}\right]\right)$$
$$\Downarrow$$

> The two objects will be separated in a $\left[\frac{U}{V}\right]$ segmentation according to case I of Eq. (2)

$$\text{Additionally, let} \quad V_1 = V_2 = U_1 (\neq U_2)$$
$$\Downarrow$$
$$rank([U_1, V_1, U_2, V_2]) = rank([U_2, V_2])$$
$$\Downarrow$$

> The two objects will be grouped together in a $[U, V]$ segmentation according to case III of Eq. (2)

Fig. 2 provides another (visual) counter example. $\Diamond$

Notes

[1] A shorter version of this paper appeared in (Machline et al., 2002)

[2] This rank rule builds upon the observations previously made by Boult & Brown (Boult and Brown, 1991).

^{3} The matrix $[G, H]$ is of size $F \times 2N$ where N (the number of pixels) is typically very large. The segmentation method proposed in (Costeira and Kanade, 1995) is suitable only for matrices with a small N, because it is applied to $\left[\frac{U}{V}\right]^T \left[\frac{U}{V}\right]$, which is an $N \times N$ matrix. The segmentation method of (Gear, 1998), however, is applied to $\left[\frac{U}{V}\right]_{2F \times N}$, and is therefore more suitable for larger N.

References

Anandan, P. and M. Irani 2002. Factorization with Uncertainty. *International Journal of Computer Vision*, 49(2-3):101–116, September, 2002.

Ayer, S. and H. Sawhney 1995. Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding. *International Conference on Computer Vision*, pp. 777–784, 1995.

Boult, T. E. and L. G. Brown 1991. Factorization-based Segmentation of Motions. *Proc. of the IEEE Workshop on Motion Understanding*, pp. 179–186, 1991.

Brand, M. 2001. Morphable 3D Models from Video. *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, volume II, pp. 456–463, 2001.

Bregler, C., A. Hertzmann and H. Biermann 2000. Recovering Non-Rigid 3D Shape from Image Streams. *IEEE Conference on Computer Vision and Pattern Recognition*, South-Carolina, volume II, pp. 690–696, 2000.

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

Gear, C. W. 1998. Multibody Grouping from Motion Images. *International Journal of Computer Vision*, 2(29):133–150, 1998.

Ichimura, N. 2000. A Robust and Efficient Motion Segmentation Based on Orthogonal Projection Matrix of Shape Space. *IEEE Conference on Computer Vision and Pattern Recognition*, South-Carolina, Volume II, pp. 446–452, 2000.

Irani, M. 2002. Multi-Frame Correspondence Estimation Using Subspace Constraints. *International Journal of Computer Vision*, 48(3):173–194, July, 2002.

Irani, M., B. Rousso and S. Peleg 1992. Detecting and Tracking Multiple Moving Objects Using Temporal Integration. *European Conference on Computer Vision*, Santa Margarita Ligure, pp. 282–287, May, 1992.

Kanatani, K. 2001. Motion segmentation by subspace separation and model selection. *International Conference on Computer Vision*, Vancouver, Canada, Volume 1, pp. 301–306, July, 2001.

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

M. Machline, L.Zelnik-Manor and M.Irani 2002. Multi-Body Segmentation: Revisiting Motion Consistency. *Workshop on Vision and Modeling of Dynamic Scene*, 2002.

Morris, D. D. and T. Kanade 1998. A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertainty Models. *International Conference on Computer Vision*, Bombay, India, pp. 696–702, 1998.

Torr, P. H. S. 1998. Geometric Motion Segmentation and Model Selection. J. Lasenby, A. Zisserman, R. Cipolla and H. Longuet-Higgins, editors, *Philosophical Transactions of the Royal Society A*, pp. 1321–1340, Roy Soc, 1998.

Torr, P. H. S. and A. Zisserman 1998. Concerning Bayesian Motion Segmentation, Model Averaging, Matching and the Trifocal Tensor. *European Conference on Computer Vision*, pp. 511–527, 1998.

Torresani, L., D. B. Yang, E. J. Alexander and C. Bregler 2001. Tracking and Modeling Non-Rigid Objects with Rank Constraints. *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, Volume I, pp. 493–500, 2001.

Wang, J. and E. Adelson 1993. Layered Representation for Motion Analysis. *IEEE Conference on Computer Vision and Pattern Recognition*, New York, pp. 361–366, June, 1993.