

Multiscale Segmentation by Combining Motion and Intensity Cues

Meirav Galun Alexander Apartsin Ronen Basri*
The Weizmann Institute of Science
Dept. of Computer Science and Applied Mathematics
Rehovot 76100, Israel

Abstract

We present a multiscale method for motion segmentation. Our method begins with local, ambiguous optical flow measurements. It uses a process of aggregation to resolve the ambiguities and reach reliable estimates of the motion. In addition, as the aggregation process proceeds and larger aggregates are identified it employs a progressively more complex model to describe the motion. In particular, we proceed by recovering translational motion at fine levels, through affine transformation at intermediate levels, to 3D motion (described by a fundamental matrix) at the coarsest levels. Finally, the method is integrated with a segmentation method that uses intensity cues. We further demonstrate the utility of the method on both random dot and real motion sequences.

1. Introduction

Segmentation of objects based on their motion is perceptually striking, as is exemplified by motion sequences containing random dots. Finding satisfactory algorithmic solutions to this problem, however, has remained a challenge. Algorithmic approaches to motion segmentation seem to face both the difficulties that complicate the task of intensity-based segmentation along with the challenges that make motion estimation hard. Issues that complicate segmentation include devising an appropriate measure of similarity and rules of clustering to correctly separate the various segments. Similarly, difficulties in motion estimation are due to the sparseness of motion cues, particularly their absence in uniform regions and due to the aperture problem. Furthermore, of crucial importance is the selection of an appropriate motion model.

A number of effective algorithms have been proposed

*Research was supported in part by the US-Israel Binational Science Foundation grant number 2002/254 and by the European Commission Project IST-2002-506766 Aim Shape. The vision group at the Weizmann Inst. is supported in part by the Moross Foundation.

to address the problem of motion segmentation, many of which produce convincing results on quite complex motion sequences. These algorithms differ in the kind of information they use (sparse features versus dense intensity information) and the motion model they impose (2D parametric versus motion in 3D). Some of these approaches also recognize the importance of combining optical flow measurements with intensity information to solve the problem of motion segmentation. Motion segmentation approaches that use dense intensity information largely impose 2D parametric motion models (mostly translation or affine). These include layered representations [19, 21] (see also [2]; also [1, 20] attempt to relax some of the main requirements of layered approaches), variational methods [3, 5], graph-cuts algorithms [6, 14], and sequential dominant motion removal [12]. Handling 3D motion is usually achieved by extracting and tracking a sparse set of features. Among these are subspace methods, which assume orthographic projection [4, 8, 11, 9] and their generalization to perspective projection [18] ([22] attempt to apply these methods directly to intensities). Other feature-based methods deal also with perspective projection [15, 16].

In this paper we describe a multiscale scheme that enables, through the use of hierarchical bottom-up processing, to overcome some of the crucial difficulties in motion segmentation. In particular, our scheme combines motion with intensity cues. The method determines the segments adaptively and estimate their motion by varying the motion model according to the amount of statistics available in each segment. We have implemented three motion models, translation, affine, and 3D rigid motion followed by perspective projection. The method we present is a two-frame approach developed to work with small motions. It relies on the weighted aggregation framework [13], which finds segments that optimize a normalized cuts measure using algebraic multigrid computation. Finally, the method is efficient, with linear runtime complexity in the number of pixels. We demonstrate the utility of our method through experiments on both random dot and real image pairs.

2. Aggregation of Motion Cues

In this section we focus our attention on motion cues alone and defer the discussion of how we integrate them with intensity-based segmentation to Section 3. The method we present performs motion segmentation by applying a sequence of *coarsening* steps, each of which further clusters the pixels in the image according to their motion, producing fewer aggregates of pixels of larger sizes. We refer to the aggregates produced by each coarsening step as a *level of scale*, with the *fine* levels containing the small aggregates produced after the first few steps of the algorithm, and the *coarse* levels containing the larger aggregates produced in subsequent steps. The primary objective of these coarsening steps is to determine which collections of pixels share a unified motion. This is achieved by simultaneously resolving motion ambiguities and describing the motion by the appropriate model. This paper considers three types of motions - translation, affine, and rigid transformation in 3D. In the future we plan to extend this by handling 2D-homographies and nonrigid transformations.

Every coarsening step is composed of two parts, clustering and re-estimation. For clustering we select a set of *seed* elements, and then associate every element from the previous level to these seeds by soft assignment. Once the clusters are determined we estimate the common motion of the cluster. Here the parameters of the motion are determined and ambiguities are resolved.

As this iterative coarsening procedure proceeds we gradually modify the model used to describe the motion of aggregates. At finer levels we seek to determine the translation of aggregates. We achieve this by applying a process of sharpening the raw motion cues. This process allow us to identify either the translation of the center of mass of an aggregate or a 1-D constraint on this motion. Later on, as sufficient translational information is accumulated, we use this information to determine more complex motions, including affine transformation and rigid motion in 3D. Below we describe the different components of the algorithm.

2.1. Initial Optical Flow Measurements

Measuring optical flow is complex, partly because local information usually is insufficient to determine the motion of a given pixel. In particular, pixels near edges are subject to a 1-D aperture ambiguity, and pixels within uniform regions are subject to a 2-D ambiguity. To represent this ambiguity we chose to follow the method of [14] and represent the initial optical flow measurements as a *motion profile*.

Let Im_1 and Im_2 denote the two input images. Using homogeneous coordinates, let $\mathbf{x}_i = (x_i, y_i, 1)^T$ denote a pixel in Im_1 . The motion profile $M_i(\mathbf{u})$ is a normalized 2-D histogram reflecting our estimate of the probability that

the optical flow at \mathbf{x}_i is given by $\mathbf{u} = (u, v, 0)^T$. To estimate this histogram we compare a 3×3 window from Im_1 centered at \mathbf{x}_i with similar windows in Im_2 centered at offsets \mathbf{u} within a bounded distance from \mathbf{x}_i . Using the SSD (sum of squares distance) between the intensities in the two windows we set the motion profile to be

$$M_i(\mathbf{u}) = \frac{1}{Z} \left(e^{-\alpha \text{SSD}(\text{Im}_1(\mathbf{x}_i), \text{Im}_2(\mathbf{x}_i + \mathbf{u}))} + C \right). \quad (1)$$

The constant α controls the penalty due to difference in intensity; assuming brightness constancy, α should be set according to the level of noise in the images. The constant term C is added to ensure that no offset is assigned zero probability (since there is always a chance that the pixel changes its intensities after motion, e.g., due to occlusion). Finally, Z is a normalizing factor set to ensure that the entries in the histogram sum to 1. We can in general use a prior to modulate this expression (e.g., incorporating small motion assumption). In our implementation we used a uniform prior, resulting in the expression (1). Another issue is how to initialize the motion profile of pixels near the boundaries of the image. Denote by k the number of cells contained in a profile, we assign $1/k$ to each cell corresponding to motion that exceeds the boundaries of the image. The rest of the cells are assigned proportionally to the expression (1). To account for light changes, SSD can be replaced by a flexible measure such as normalized correlation.

2.2. Optical Flow Disambiguation

To handle translation, we make the simplifying assumption that pixels provide independent information about their motion and use this assumption to evaluate which pixels should cluster together. According to this independence assumption, the joint probability that two pixels \mathbf{x}_i and \mathbf{x}_j with motion profiles M_i and M_j share a common translation \mathbf{u} is given by $M_i(\mathbf{u})M_j(\mathbf{u})$. In reality, we may want to cluster together neighboring pixels that share a common motion even if this motion is non-translational (e.g., rotation). To account for small deviations from translation, and to account for noise, we first smooth the two motion profiles before taking their product. Denote by $g(M(\mathbf{u})) = G * M(\mathbf{u})$, where G denotes a Gaussian function with zero mean and a small standard deviation σ (we used $\sigma = 0.5$), and $*$ denotes convolution. Then the chance that two pixels share roughly the same translation is given by

$$g(M_i(\mathbf{u}))g(M_j(\mathbf{u})). \quad (2)$$

To evaluate the resemblance of the motion profiles of two neighboring pixels we follow [14] and define a measure based on the normalized correlation between the profiles. Define the distance between two profiles as $d_{\text{profile}} =$

$1 - \sum_{\mathbf{u}} g(M_i(\mathbf{u}))g(M_j(\mathbf{u}))$ and the similarity as

$$w_{ij} = e^{-\beta d_{\text{profile}}}, \quad (3)$$

where β is a scaling factor. At the finest level each pixel is connected to its four immediate neighbors with $w_{ij} > 0$, and every furthest pairs satisfy $w_{ij} = 0$.

Each coarsening step begins by selecting a subset of the elements from the previous level (pixels in the finest level, aggregates of pixels in higher levels) as *seeds*, with the constraint that all other elements are strongly associated with (subsets of) these seeds (using the similarity in (3) as a measure of association). We further prescribe an association weight p_{ik} to express the strength of association of a finer element i to a coarser seed k :

$$p_{ik} = \frac{w_{ik}}{\sum_l w_{il}}, \quad (4)$$

where the sum runs over all seeds. The values p_{ik} are non-zero only in a close neighborhood of each element.

Once all the association weights are determined we can construct a common motion profile for the new aggregates. By generalizing (2) we obtain

$$M_k(\mathbf{u}) = \frac{1}{Z} \prod_i g(M_i(\mathbf{u}))^{p_{ik} V_i / \bar{V}}. \quad (5)$$

According to this expression the motion profile of an aggregate k is given by the product of all the motion profiles of its children, where the power weighs each term according to the strength of association of a child to the seed and accounts for its volume (with V_i the volume of child i and \bar{V} the average volume of an aggregate in its level). Z is the appropriate normalizing constant. With this formula the motion profile of a pixel is distributed between all the seeds it is associated with.

This coarsening process, which is composed of seed selection, associating elements with the seeds, and computing the new motion profiles, is repeated, creating at every subsequent level fewer aggregates of larger sizes. Expressing the motion profile of these aggregates as a product of the profiles of their children results in a sharp decrease of the probabilities of incompatible motions relative to that of the correct translation. In textured regions it is often sufficient to perform one or two coarsening steps to obtain a sharply peaked motion profile. In contrast, motion profiles within uniform regions usually remain ambiguous until either the aggregation process combines them with areas that contain features or the entire uniform region is clustered. Combining this process with intensity based segmentation (Section 3.1) ensures the clustering of both uniform regions and texture features, and this in turn assists in obtaining meaningful optical flow measurements.

During this aggregation process we examine the motion profile of each of the aggregates in each level to determine

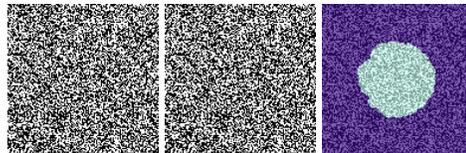


Figure 1. From left to right: a random dot pair containing a disc rotating by 6 degrees and a background translating by 5 pixels and segmentation results obtained with our method (displayed by a color overlay on the top of the leftmost picture).

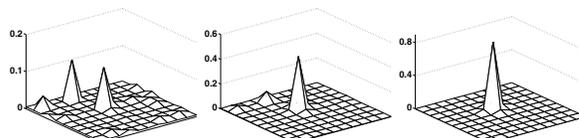


Figure 2. Resolving optical flow ambiguity. From left to right: motion profiles obtained at scales 0, 1, and 2.

if its profile is sharply peaked, in which case we label the aggregate as *peaked* and set the optical flow for the center of the aggregate according to the location of the peak. If a profile is not sharply peaked we further examine it to test whether it contains a line with elevated probability values. In that case we conclude that the aggregate is subject to an aperture problem. We thus label the aggregate as *bar-peaked* and associate a normal flow vector to the center of the aggregate according to the parameters of this line.

To illustrate the motion disambiguation process we ran the method on a random dot image pair containing a disc rotating by 6 degrees and a background translating by 5 pixels (Fig. 1). In Fig. 2 we show an example of the progression of the motion profile with scale at a certain image location. Notice the ambiguous motion profile at level 0 which is resolved in the next two levels up, yielding a peaked profile.

2.3. Affine Transformation

As we proceed with the aggregation process, the size of aggregates increases, and translation ceases to accurately reflect their motion. We therefore wish to use a more complex model to describe this motion. We do so by fitting an affine transformation for each aggregate. Unfortunately, the motion profile of an aggregate cannot be used to determine its affine motion since it does not contain sufficient degrees of freedom. Instead, we accumulate constraints from the peaked and bar-peaked sub-aggregates, and use these constraints to determine the affine motion.

Specifically, suppose the flow at a point \mathbf{x}_i is given by \mathbf{u} . We wish to fit a 3×3 matrix A that satisfies $A\mathbf{x}_i = \mathbf{u}$ (with the last row of A containing $(0, 0, 0)$). There are two weights we need to take into account. The first is the degree

to which \mathbf{x}_i (which denotes the center of mass of some sub-aggregate i a few levels down) belongs to the aggregate k for which we perform the computation. Generalizing (4) to i and k separated by any number of levels, we denote this weight by p_{ik} . The second weight reflects our belief in the optical flow measurement, as is expressed in the motion profile, given by $M_i(\mathbf{u})$. Incorporating these weights we seek a matrix A that minimizes

$$\min_A \sum_i \sum_{\mathbf{u}} p_{ik} M_i(\mathbf{u}) \|A\mathbf{x}_i - \mathbf{u}\|^2, \quad (6)$$

with the summation going over all sub-aggregates i and their motion profiles \mathbf{u} . Taking derivatives with respect to A we obtain¹

$$A \left(\sum_i p_{ik} \mathbf{x}_i \mathbf{x}_i^T \right) = \sum_i \sum_{\mathbf{u}} p_{ik} M_i(\mathbf{u}) \mathbf{u} \mathbf{x}_i^T. \quad (7)$$

This provides a set of six equations in the six unknown components of A , which uses moments of points up to second order (left hand side) and bilinear moments of their motion (right hand side). We collect these moments from the peaked and bar-peaked sub-aggregates of all finer levels using their motion profiles.

To obtain these moments we apply a process of accumulation in which we use the moments computed at every level to compute the moments of the subsequent level. A straightforward accumulation of moments may result in bias, as the motion profile can suffer from noise or the motion profile may still be ambiguous. We therefore apply a *selective* moment aggregation in a way that only peaked or bar-peaked sub-aggregates contribute to the moment accumulation. We label an aggregate as peaked (or bar-peaked) by *heredity* if at least one of its strongly-related children is labelled peaked. In this case we compute its moments as a weighted sum of the moments of its peaked (or bar-peaked) children. If an aggregate is not labelled peaked (or bar-peaked) by heredity we further examine if most of the energy in its motion profile is concentrated around a single location (or a line), in which case we label the aggregate as peaked (respectively bar-peaked) and initialize its moments using the following expression:

$$\sum_{\mathbf{u}} M_k(\mathbf{u}) x^{\delta_1} y^{\delta_2} u^{\delta_3} v^{\delta_4}, \quad (8)$$

where (x, y) are the center of mass of the aggregate, $\delta_j \geq 0$ are integers and $\sum \delta_j \leq 2$. Note that the moments accumulated this way adaptively collect information from aggregates of different scales.

The zeroth order moment indicates the number of points contributing to the moments. Since a peaked aggregate contributes two constraints and a bar-peaked contributes one

¹This and subsequent derivations can be obtained using $\partial \mathbf{y}^T A \mathbf{x} / \partial A = \mathbf{y} \mathbf{x}^T$, and $\partial \mathbf{y}^T A^T A \mathbf{x} / \partial A = A(\mathbf{y} \mathbf{x}^T + \mathbf{x} \mathbf{y}^T)$.

constraint, we can use the zeroth order moment to determine if a sufficient number of points has been identified to determine an affine transformation. Whenever we detect aggregates for which there are no sufficient constraints to determine an affine transformation we assign to them the identity matrix for the linear part and translation according to the constraints available. If no constraints are available we consider these aggregates as stationary.

Once we describe the motion of aggregates by an affine transformation further coarsening requires us to compare these motions. A simple way to compare affine transformations is by directly comparing their components. However, a significant difference in the components of a transformation may not necessarily imply a similar difference in the effect of the transformation. To account for this we compare two affine transformations by the difference between the motion they induce on the relevant aggregates. We use a top-down process in which we examine sub-aggregates at two finer levels down. Denote by A_k the affine transformation of aggregate k , and the center of mass of its sub-aggregates (two levels down) by $\mathbf{x}_i = (x_i, y_i, 1)^T$ and their respective association weights to k by p_{ik} . The (non-symmetric) difference between the affine transformations assigned to aggregates k and l , d_{kl} , is defined as

$$d_{kl} = \left(\frac{\sum_i p_{ik} (A_k \mathbf{x}_i - A_l \mathbf{x}_i)^2}{\sum_i p_{ik}} \right)^{\frac{1}{2}}. \quad (9)$$

Similarly, the difference in the other direction d_{lk} is calculated. The joint affine transformation distance between the aggregates k and l is $d_{\text{affine}} = \min(d_{kl}, d_{lk})$. From a certain level on we substitute d_{profile} in (3) by this measure.

An illustrative example is provided in Fig. 3. We use the same random dot pair presented in Fig. 1. The figure shows aggregates obtained at level 6 and the different motions associated with them. On the left we show the translation of each aggregate, as it is determined by the motion profile. On the right we show the rotation angle of each aggregate as it is determined by the affine model. (For the background aggregates this angle is nearly zero.) It can be seen that all the aggregates composing the disc have similar rotation angle. They were thus all aggregated at the coarsest level to form a single segment (Fig. 1, right). The rotation angle estimated for the entire disc at this level was 6.0007 degrees. Note that pixels at the bottom of the disc are subject to a motion that is very similar to the motion of the background, yet they were quite accurately segmented due to the optimization criteria dictated by the normalized cut criterion. In contrast, the left side of the disc was ruggedly segmented because background pixels near this boundary underwent occlusion in the second image.

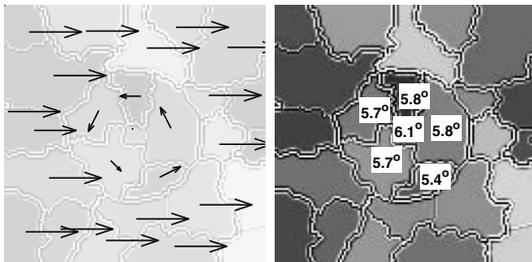


Figure 3. From translation model to affine model.

2.4. Fundamental Matrix

Video sequences are often taken with the camera moving. Generically, such a motion produces perspective distortions throughout the image, making it difficult to separate moving objects from a stationary background. To account for this we compute for each aggregate at the top-most levels a fundamental matrix and compare the obtained matrices in an attempt to cluster together segments describing stationary portions of the scene. Below we describe how this fundamental matrix is computed and compared.

Using the same notation introduced in the previous section we seek a 3×3 rank 2 matrix F that minimizes

$$\min_F \sum_i \sum_{\mathbf{u}} p_{ik} M_i(\mathbf{u}) ((\mathbf{x}_i + \mathbf{u})^T F \mathbf{x}_i)^2. \quad (10)$$

Taking derivatives with respect to F we obtain

$$\sum_i \sum_{\mathbf{u}} p_{ik} M_i(\mathbf{u}) ((\mathbf{x}_i + \mathbf{u})^T F \mathbf{x}_i) (\mathbf{x}_i + \mathbf{u}) \mathbf{x}_i^T = 0. \quad (11)$$

This provides a set of nine homogeneous equations in the components of F , which uses motion moments of points up to fourth order (defined as in (8) with $0 \leq \delta_j \leq 4$). We collect these moments from the peaked sub-aggregates only. (The bar-peaked sub-aggregates generally do not constrain the fundamental matrix since the constraint line may intersect many epipolar constraints.) We solve the equation using the eight point algorithm using the normalization procedure proposed by Hartley [10] followed by rank reduction. Again, we use the zeroth order moment to determine whether we accumulated sufficient equations for a solution. Degeneracies can be handled as in [17], although this has not yet been implemented in our method.

The calculation of the fundamental matrix is followed by a comparison of the fundamental matrices between each two neighboring aggregates. In this case it is not straightforward to apply the same comparison procedure used in the affine case, since a fundamental matrix provides only a line constraint on the location of each point. We therefore chose to use the simpler method of comparing the entries of the two matrices using an l_2 norm. The resulting measure,

Motion Segmentation:

- Given two images Im_1 and Im_2 , prepare for each pixel in Im_1 a motion profile (1).
- Assign a weight to each pair of neighboring pixels according to the normalized correlation between their motion profiles (3).

Coarsening iteration:

1. Clustering:

- (a) Select a set of seeds such that the remaining elements are strongly connected to this set.
- (b) Define the strength of association of a fine element i to a coarse seed k (4).

2. Re-estimation: For each seed

- (a) Calculate the motion profile of the seed by multiplying the profiles of its children (5).
- (b) Examine whether the seed is peaked, by heredity or by itself.
- (c) If it is not peaked check if it is bar-peaked, by heredity or by itself.
- (d) Accumulate adaptively, moments (orders one to four) originated by peaked seeds.
- (e) Accumulate separately, moments (orders one and two) originated by bar-peaked seeds.
- (f) If there is enough statistics, calculate affine transformation by merging moments from peaked and bar-peaked profiles.
- (g) If there is enough statistics, calculate fundamental matrix from peaked profiles.

3. Calculate for each neighboring seeds cross correlation distance, affine transform distance and fundamental matrix distance.

4. Modify appropriately the similarities between neighboring seeds.

Table 1. Outline of the motion segmentation algorithm

$d_{\text{fundamental}}$ is then use to replace d_{profile} in (3). An outline of our motion segmentation algorithm is provided in Table 1.

3. Using Intensity Cues

To combine motion with intensity cues we integrate our motion segmentation algorithm with the Segmentation by Weighted Aggregation (SWA) algorithm [13]. This algorithm has been extended to also handle texture cues [7], although our implementation did not make use of these cues. Below we describe the main principles behind the SWA algorithm (Section 3.1) and how we combine motion with intensity cues in this framework (Section 3.2).

3.1. SWA Segmentation

The SWA algorithm is a multiscale graph partitioning algorithm. Given an image, it constructs a graph $G =$

(V, W) , with nodes in V representing image pixels and the symmetric edge weight matrix W representing the affinities between neighboring pixels. To evaluate segments it defines a saliency measure as follows. Every node v_i , ($1 \leq i \leq N$, where $N = \|V\|$) is associated with a state variable u_i , and every candidate segment $S = \{v_1, v_2, \dots, v_m\} \subseteq V$ is associated with a state $u = (u_1, u_2, \dots, u_N)$ such that

$$u_i = \begin{cases} 1 & \text{if } v_i \in S \\ 0 & \text{if } v_i \notin S. \end{cases} \quad (12)$$

The saliency associated with S is defined by

$$\Gamma(S) = \frac{u^T L u}{\frac{1}{2} u^T W u}, \quad (13)$$

where L is the Laplacian matrix whose elements are

$$l_{ij} = \begin{cases} \sum_{k(k \neq i)} w_{ik} & i = j \\ -w_{ij} & i \neq j. \end{cases} \quad (14)$$

This saliency measure sums the weights along the boundaries of S normalized by the internal weights. Segments that yield small values of $\Gamma(S)$ are considered salient. Allowing arbitrary real assignments to u the minimum for Γ is obtained by the minimal generalized eigenvector u of $L u = \lambda W u$, with the condition that $\lambda > 0$.

The SWA algorithm finds the best partitions (0-1 assignments of u) by recursively producing smaller representative sets of seeds, such that every pixel is strongly-connected to the set of seeds. Denote by $U = (U_1, U_2, \dots, U_n)$ the coarse level state vector. We construct a sparse, $N \times n$ matrix P such that $u \approx P U$. P is called the inter-scale interpolation matrix. Using this matrix the saliency measure Γ can be written as

$$\Gamma = \frac{u^T L u}{\frac{1}{2} u^T W u} \approx \frac{U^T P^T L P U}{\frac{1}{2} U^T P^T W P U}. \quad (15)$$

To calculate the right hand side of this equation we need to compute the matrix $P^T W P$, which is a coarse representation of the original weight matrix. This is called *weighted aggregation*. Exploiting the sparseness of P , this product is computed in linear time.

3.2. Combining Motion with Intensities

In this section we will attach the superscripts M and I to denote measures corresponding to motion and intensity information respectively. The multiscale partitioning procedure described in the previous section can be used for segmentation combining motion and intensity cues in the following way. Given a pair of images, we begin by constructing a 4-connected graph $G = (V, W)$, where every pixel is represented by a node $v_i \in V$, and every pair of

neighboring pixels are connected with an edge with weight w_{ij} . This weight is a product of two term. A measure reflecting the contrast between the two pixels i and j in Im_1

$$w_{ij}^I = e^{-\tilde{\beta}|I_i - I_j|}, \quad (16)$$

where I_i and I_j denote the intensities of the two neighboring pixels, $\tilde{\beta}$ is a positive constant, and a measure w_{ij}^M reflecting the difference in the motion profiles associated to the two pixels (3).

At each coarsening step, we first determine the next coarser graph using the weighted aggregation procedure. This will produce a graph that is roughly equivalent to the finer level graph, with weights inherited from the previous level. We then modify the weights in the new graph to incorporate coarser measures of differences between neighboring aggregates. Specifically, for every two aggregates we multiply these weights by the a measure of difference between their average intensities and possibly their variance (of the form similar to (16)), and likewise by a measure reflecting the difference between their motion profiles (3), and at higher scales the difference between their affine transformations and fundamental matrices.

4 Experiments

We implemented the combined algorithm and applied it to a collection of image pairs. (The image pairs can be seen in motion in the supplementary file.) We set the parameters around the following values: $\alpha = 10$, $\beta = 4$, $\tilde{\beta} = 7$. The motion profile distance d_{profile} was evaluated from the finest scale to scale 4. The affine distance d_{affine} was evaluated from a scale 5, and $d_{\text{fundamental}}$ was applied at the two topmost levels. We label a motion profile as peaked if the volume below a 3×3 window around the maximum of the profile exceeds 0.5 and as bar-peaked if the volume around a line through the profile exceeds 0.8. Our non-optimized implementation runs in less than 10 seconds on an image pair with 200×250 pixels on a Pentium 4 PC. To demonstrate the handling of motion cues alone we first applied the algorithm to several pairs of images containing a moving sequence of random dots. Figure 4 shows a random dot sequence containing a pair of translating squares on a stationary background along with segmentation results (displayed by a color overlay on top of the leftmost picture) and motion vectors computed by our algorithm. In this and other examples we extracted the motion vectors identified in peaked aggregates at levels 4-5. The bar-peaked aggregates are not displayed in these images.

Figure 5 contains a random dot sequence of a foreground sphere placed in front of a three wall background. Both the foreground and the background are moving in two separate 3D motions. The use of fundamental matrices was critical in this case for obtaining the three walls in one piece.

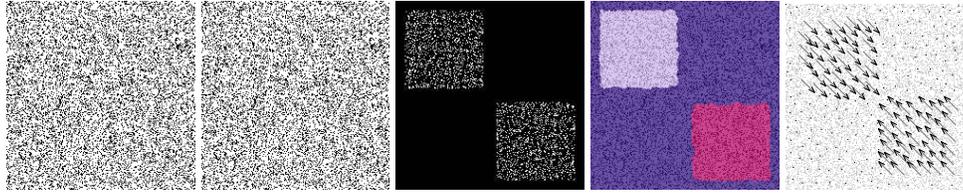


Figure 4. From left to right: a random dot pair containing two translating squares, a difference image, segmentation results obtained with our method, and motion vectors obtained from peaked aggregates at levels 4-5.

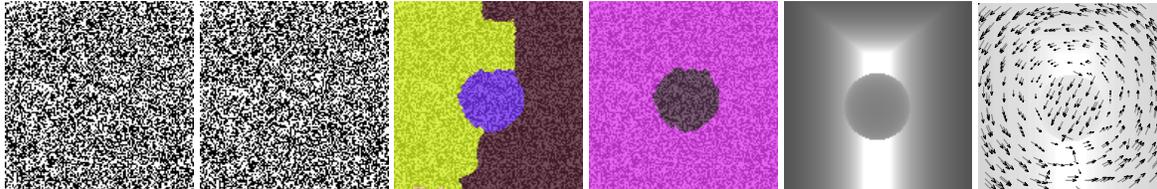


Figure 5. A random dot pair containing foreground sphere, and background walls, with different motions in 3D. Results obtained by applying our method with affine transformation only, results obtained by applying a comparison of fundamental matrices at the coarsest levels, depth image (intensity proportional to distance from camera), and motion vectors displayed on the depth image.

The rest of the figures show the results obtained with our method on real motion pairs. Figure 6 shows a car exiting to a street with some camera motion. Using intensity alone (SWA algorithm) resulted in attaching the dark top of the car with the background. Using also motion cues the car is segmented in one piece (although the wheels are incorrectly attached to the background). The figure further shows the epipolar lines computed with our method for the background segment. In Figure 7 the arm of the octopus is connected in one piece despite a dark section in the middle mainly because of continuity in motion. Similar results are obtained in Figure 8.

5 Conclusion

We have presented an efficient multiscale algorithm for image segmentation that combines motion with intensity cues. The algorithm uses bottom-up processing to disambiguate motion measurements and determine an appropriate motion model for various regions in the image. We have demonstrated the algorithm by applying it to a variety of random dot and real motion pairs.

Our algorithm is related to several existing algorithms. We represent initial motion measurements in the form of motion profiles and apply a graph algorithm to find minimal cuts in the graph (as in [14]) using the algorithm proposed in [13]. In addition, similar to layer approaches [19, 21, 1, 20] our algorithm is composed of a sequence of clustering and re-estimation steps. However, unlike these methods, our method uses coarsening steps to disambiguate motion measurements and to adaptively select a motion model according to the amount of statistics available. Because each

coarsening step reduces the number of clusters handled by the algorithm, the cost is linear in the number of pixels.

In the future we plan to extend the method to handle sequences composed of three or more frames and expand the repertoire of motion models by incorporating 2D homographies and nonrigid transformations. Also of importance is to handle the “edge assignment” problem in order to determine for boundary edges to which segment their motion is relevant. This is important particularly in uniform regions when there exist no supporting features in addition to the boundary edges to determine the motion of a segment. In addition, we wish to exploit the multiscale nature of our algorithm to relax the assumption of small motion. Finally, we plan to explore learning approaches to automatically set the parameters of this process.

References

- [1] S. Ayer, H.S. Sawhney, Layered Representation of Motion Video Using Robust Maximum-Likelihood Estimation of Mixture Models and MDL Encoding, *ICCV*: 777-784, 1995.
- [2] M.J. Black, A. Jepson, Estimating optical flow in segmented images using variable-order parametric models with local deformations, *PAMI*, **18**(10): 972-986, 1996.
- [3] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, *ECCV*: 25-36, 2004.
- [4] J. Costeira and T. Kanade, A Multi-body Factorization Method for Motion Analysis, *IJCV* **29**(3): 159-179, 1998.
- [5] D. Cremers, A Variational Framework for Image Segmentation Combining Motion Estimation and Shape Regularization, *CVPR I*: 53-58, 2003.
- [6] C. Fowlkes, S. Belongie, F. Chung, J. Malik. Spectral Grouping using the nystrom method”, *PAMI*, **26**(2): 214-225, 2004.



Figure 6. Top row: a motion pair, a difference image, results obtained by applying segmentation based on intensity cues alone. Bottom row: results obtained by combining motion with intensity cues (our method), motion vectors, and epipolar lines computed for the background segment.

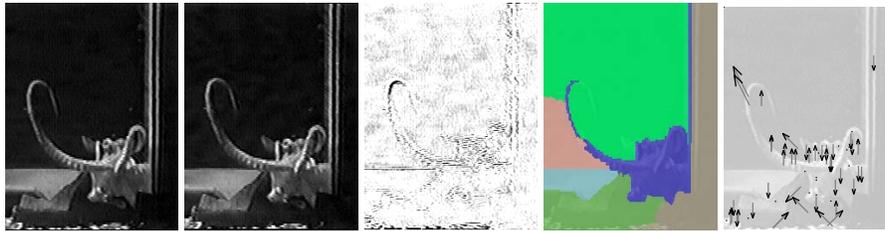


Figure 7. A motion pair (courtesy of Yoram Yekutieli of the interdisciplinary center for neural computation at the Hebrew University), a difference image, results obtained by applying our segmentation method, and motion vectors.

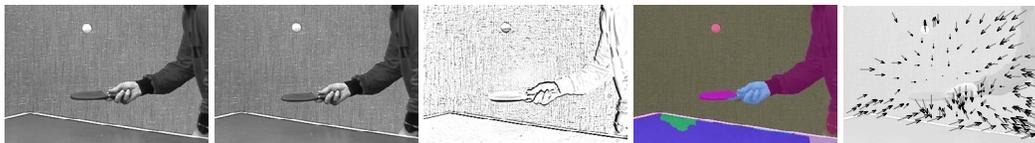


Figure 8. A motion pair, a difference image, results obtained by applying our segmentation method, and motion vectors.

- [7] M. Galun, E. Sharon, R. Basri, A. Brandt, Texture segmentation by multiscale aggregation of filter responses and shape elements, *ICCV*: 716–723, 2003.
- [8] C.W. Gear, Multibody grouping from motion images, *IJCV*, **2**(29): 133–150, 1998.
- [9] A. Gruber, Y. Weiss, Multibody factorization with uncertainty and missing data using the EM algorithm, *CVPR*, **1**: 707–714, 2004.
- [10] R.I. Hartley, In Defense of the Eight-Point Algorithm, *PAMI*, **19**(6): 580–593, 1997.
- [11] K. Kanatani, Evaluation and selection of models for motion segmentation. *ECCV*: 335–349, 2002.
- [12] M. Irani, B. Rousso, S. Peleg, Detecting and tracking multiple moving objects using temporal integration, *ECCV*: 282–287, 1992.
- [13] E. Sharon, A. Brandt, R. Basri, Segmentation and boundary detection using multiscale intensity measurements, *CVPR*, **I**:469–476, 2001.
- [14] J. Shi, J. Malik, Motion segmentation and tracking using normalized cuts” *ICCV*: 1154–1160, 1998.
- [15] P.H.S. Torr, A.W. Fitzgibbon, A. Zisserman, Maintaining Multiple Motion Model Hypotheses Over Many Views to Recover Matching and Structure, *ICCV*: 485–491 1998.
- [16] W.S. Tong, C.K. Tang, G. Medioni, Simultaneous Epipolar Geometry Estimation and Motion Segmentation by 4D Tensor Voting in Joint Image Space, *PAMI*, **26**(9), 1167–1184, 2004.
- [17] P.H.S. Torr, A. Zisserman, S. Maybank, Robust Detection of Degenerate Configurations for the Fundamental Matrix, *CVIU*, **71**(3): 312–333, 1998.
- [18] R. Vidal, S. Soatto, Y. Ma and S. Sastry, Segmentation of Dynamic Scenes from the Multibody Fundamental Matrix, *ECCV Workshop on Vision and Modeling of Dynamic Scenes*, 2002.
- [19] J.Y.A. Wang, E.H. Adelson, Representing Moving Images with Layers. *IEEE Trans. on Image Processing*, **3**(5):625–638, 1994.
- [20] Y. Weiss, Smoothness in Layers: Motion segmentation using non-parametric mixture estimation. *CVPR*, 520–527, 1997.
- [21] Y. Weiss, E.H. Adelson, A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. *CVPR*, 321–326, 1996.
- [22] L. Zelnik-Manor, M. Machline, M. Irani, Multi-Body factorization with uncertainty: revisiting motion consistency, *IJCV*, 2004.