IMAGE AND VIDEO SEGMENTATION: THE NORMALIZED CUT FRAMEWORK

Jianbo Shi, Serge Belongie, Thomas Leung and Jitendra Malik

Computer Science Division, University of California at Berkeley, Berkeley, CA 94720
{jshi,sjb,leungt,malik}@cs.berkeley.edu

ABSTRACT

In this paper, we propose a segmentation system based on the normalized cut framework proposed by Shi and Malik (1997). The goal is to partition the image from a big picture point of view. Perceptually significant groups are detected first while small variations and details are treated later. Different image features — intensity, color, texture, contour continuity, motion and stereo disparity are treated in one uniform framework.

1. INTRODUCTION

There has been a large amount of work on image and video segmentation. We will review some representative examples here. The most widely used segmentation algorithm is edge detection [3]. An edge detector marks all the pixels where there are big discontinuities in intensity, color or texture. The cue of contour continuity is exploited to link the edgels together to form long contours [8]. Texture information is encoded as the responses to a set of linear filters [5]. Another formulation for segmentation is the *variational formulation*. Pixel similarities are defined locally, but the final segmentation is obtained by optimizing a global functional [7]. For motion segmentation, one popular algorithm is the motion layer approach — the goal is to simultaneously estimate multiple global motion models and their spatial supports. The Expectation-Maximization (EM) algorithm allows one to achieve this goal by a gradient descent search [4].

Despite all the work above, there is no satisfactory solution to image segmentation for natural scenes. The key issues pertain to natural texture, weak contrast edges and combining different cues together. In this paper, we propose a segmentation system which addresses these problems. Our system is based on the normalized cut framework proposed by Shi and Malik [10]. The goal is to partition the image from a "big picture" point of view. Perceptually significant groups are detected first while small variations and details are treated later. Different image features — intensity, color, texture, contour continuity, motion and stereo disparity are treated in one uniform framework.

2. SEGMENTATION USING NORMALIZED CUTS

In this section, we review the normalized cut framework for grouping proposed by Shi and Malik in [10]. Shi and Malik formulate visual grouping as a graph partitioning problem. The nodes of the graph are the entities that we want to partition, for example, in image segmentation, they will be the pixels; in video segmentation, they will be a space-time triplet. The edges between two nodes correspond to the *strength* with which these two nodes belong to one group, again in image segmentation, the edges of the graph will correspond to how much two pixels agree in intensity, color, etc; while in motion segmentation, the edges describe the similarity of the motion. Intuitively, the criterion for partitioning the graph will be to minimize the sum of weights of connections *across* the groups and maximize the sum of weights of connections *within* the groups.

Let $\boldsymbol{G} = \{\boldsymbol{V}, \boldsymbol{E}\}$ be a weighted undirected graph, where $\boldsymbol{V}$ are the nodes and $\boldsymbol{E}$ are the edges. Let $\boldsymbol{A}, \boldsymbol{B}$ be a partition of the graph: $\boldsymbol{A} \cup \boldsymbol{B} = \boldsymbol{V}, \boldsymbol{A} \cap \boldsymbol{B} = \emptyset$. In graph theoretic language, the similarity between these two groups is called the *cut*:

$$cut(\boldsymbol{A}, \boldsymbol{B}) = \sum_{u \in A, v \in B} w(u, v)$$

where $w(u, v)$ is the weight on the edge between nodes u and v. Shi and Malik proposed to use a *normalized* similarity criterion to evaluate a partition. They call it the *normalized cut*:

$$Ncut(\boldsymbol{A}, \boldsymbol{B}) = \frac{cut(\boldsymbol{A}, \boldsymbol{B})}{asso(\boldsymbol{A}, \boldsymbol{V})} + \frac{cut(\boldsymbol{B}, \boldsymbol{A})}{asso(\boldsymbol{B}, \boldsymbol{V})}$$

where $asso(\boldsymbol{A}, \boldsymbol{V}) = \sum_{u \in \boldsymbol{A}, t \in \boldsymbol{V}} w(u, t)$ is the total connection from nodes in $\boldsymbol{A}$ to all the nodes in the graph. For more discussion on this criterion, please refer to [10].

One key advantage of using the normalized cut is that a good approximation to the optimal partition can be computed very efficiently. [1] Let $\boldsymbol{W}$ be the association matrix, i.e. W_{ij} is the weight between nodes i and j in the graph. Let $\boldsymbol{D}$ be the diagonal matrix such that $D_{ii} = \sum_j W_{ij}$, i.e. D_{ii} is the sum of the weights of all the connections to node i. Shi and Malik showed that the optimal partition can be found by computing:

$$\begin{aligned} \boldsymbol{y} &= \arg\min Ncut \\ &= \arg\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}} \end{aligned} \quad (1)$$

where $\boldsymbol{y} = \{a, b\}^N$ is a binary indicator vector specifying the group identity for each pixel, i.e. $y_i = a$ if pixel i belongs to group $\boldsymbol{A}$ and $y_j = b$ if pixel j belongs to $\boldsymbol{B}$. N is the number of pixels. Notice that the above expression is the Rayleigh quotient. If we relax $\boldsymbol{y}$ to take on real values (instead of two discrete values), we can optimize Equation 1 by solving a generalized eigenvalue system. Efficient algorithms with polynomial running time are well-known for solving such problems. Therefore, we can compute an approximation to the optimal partition very efficiently. For details of the derivation of Equation 1, please refer to [10].

[1] Finding the true optimal partition is an NP-complete problem.

0-8186-8821-1/98 $10.00

3. THE MASS-SPRING ANALOGY

As we have just seen, the Normalized Cut algorithm requires the solution of a generalized eigensystem involving the weighted adjacency matrix. In this section, we develop the intuition behind this process by considering a physical interpretation of the eigensystem as a mass-spring system.

One can readily verify that the symmetric positive semidefinite matrix $(\mathbf{D} - \mathbf{W})$, known in graph theory as the *Laplacian* of the graph $\mathbf{G}$, corresponds to a *stiffness matrix* while the diagonal positive semidefinite matrix $\mathbf{D}$ represents a *mass matrix*. These matrices are typically denoted by $\mathbf{K}$ and $\mathbf{M}$, respectively, and appear in the equations of motion as

$$\mathbf{M}\ddot{\mathbf{x}}(t) = -\mathbf{K}\mathbf{x}(t)$$

If we assume a solution of the form $\mathbf{x}(t) = \mathbf{v}_k \cos(\omega_k t + \phi)$, we obtain the following generalized eigenvalue problem for the time-independent part,

$$\mathbf{K}\mathbf{v}_k = \omega_k^2 \mathbf{M}\mathbf{v}_k$$

in analogy to Equation (1).

The intuition is that each pixel represents a mass and each connection weight represents a Hooke spring constant. If the system is shaken, tightly connected groups of pixels will tend to shake together.

In light of this connection, the generalized eigenvectors in Equation (1) represent normal modes of vibration of an equivalent mass-spring system based on the pairwise pixel similarities.[2] For illustrative purposes, a few normal modes for the landscape test image in Figure 2 are shown in Figure 1, together with a snapshot of a superposition of the modes.

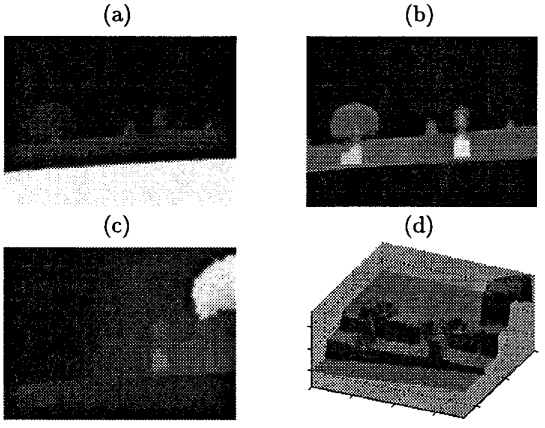

Figure 1: Three generalized eigenvectors ($\mathbf{v}_2$, $\mathbf{v}_3$ and $\mathbf{v}_4$) for the landscape test image are shown in (a)-(c). As an illustration of the connection between Normalized Cuts and the analysis of mass-spring systems, a superposition of the modes at an arbitrary time instant is shown in (d) as a surface plot.

[2]Note that since we assume free boundary conditions around the edges of the image, we ignore the first mode since it corresponds to uniform translation.

Using the mass-spring analogy, one can proceed to define a measure of similarity within the space of modes by considering the maximum extension of each spring over all time. We refer to this as the *inter-group distance*. As described in [2], the inter-group distance between two pixels i and j may be defined as the following weighted L_1 norm:

$$d_{IG}(i,j) = \sum_{k=2}^{K} \frac{1}{\omega_k} |\mathbf{v}_k^i - \mathbf{v}_k^j| \qquad (2)$$

Since the springs have large extensions between groups and small extensions within groups, an obvious application of the inter-group distance is to define a measure of local "edginess" at each pixel. Please refer to [2] for a more detailed discussion of this idea.

4. LOCAL IMAGE FEATURES

In region-based segmentation algorithms, similarity between pixels are encoded locally and there is a global routine that makes the decision of partitioning. In the normalized cut framework, local pixel similarities are encoded in the *weight* matrix $\boldsymbol{W}$ discussed in section 2. In this section, we will describe how local pixel similarities are encoded to take into account the factors of similarity in intensity, color, texture; contour continuity and common motion (or common disparity in stereopsis).

4.1. Brightness, Color and Texture

We first look at how we measure pixel similarities due to brightness, color and texture. Texture information is measured as the responses to a set of zero-mean difference of Gaussian (DOG) and difference of offset Gaussian (DOOG) kernels, similar to those used for texture analysis in [5]. We call the vector of filter responses the texture feature vector: $\boldsymbol{u}_{tex} = (f_1 * I, f_2 * I, \ldots, f_N * I)^T$. Intensity and color are measured using histograms with soft binning. We write the intensity/color feature vector as $\boldsymbol{u}_{col}$. The combined texture and intensity/color feature vector at pixel i is thus given by: $\boldsymbol{u}_i = (\boldsymbol{u}_{tex}^T, \boldsymbol{u}_{col}^T)^T$. This feature vector is normalized to have L_2 norm equal to 1: $\hat{\boldsymbol{u}}_i = \boldsymbol{u}_i / \|\boldsymbol{u}_i\|$. Notice that $\|\boldsymbol{u}_{col}\|$ is approximatly equal to a constant. The normalization step can then be seen as a form of gain control, which diminishes the contribution of the intensity/color components when there is a lot of activity in the texture components. The dissimilarity between two pixels is then defined as:

$$d_{tex,col} = (\hat{\boldsymbol{u}}_i - \hat{\boldsymbol{u}}_j)^T \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{u}}_i - \hat{\boldsymbol{u}}_j)$$

Since we are measuring texture properties at a point, the texture feature vectors are different at the center of a texel than in the space between two texels. We introduce the idea of *area completion*, which modifies the dissimilarity measure to essentially specify that the space between two texels on a texture belongs to the same surface as well. We would like to emphasize that using texture point properties together with area completion is better than the traditional way of averaging texture features in a large window, because it can handle boundaries better.

4.2. Contour Continuity

Information about curvilinear continuity can also be incorporated into the similarity measure between two pixels. Contour information can be computed "softly" through *orientation energy* [6] ($OE(x)$). Orientation energy is strong

at an extended contour of sharp contrast, while it will be weak at low contrast gaps along the contour. We enhance the orientation energy at low contrast gaps by propagating the energy from neighboring pixels along an extended contour. The probability of propagation is derived from the energy of the *elastica* curve completion model [12]. Orientation energy, after propagation, provides us with soft information about the presence of contours. Intuitively, the factor of curvilinear continuity says that two pixels belong to two different groups if there is a contour separating them. The dissimilarity is stronger if the contour is extended. Orientation energy allows us to capture this notion very easily. Given pixels p_1 and p_2, dissimilarity between them is high if the orientation energy along the line joining them is strong. Thus, if l is the straight line between p_1 and p_2 and x is a pixel on l, we define the dissimilarity due to contour continuity as:

$$d_{edg}(p_1, p_2) = \max_{x \in l}\{OE(x) - 0.5(OE(p_1) + OE(p_2))\}$$

As an alternative to this definition, one can restrict the evaluation of the orientation energy to points lying on edge contours. The edge contours can be detected and localized using, for example, zero crossings of the oriented second derivative [9]. Such a definition leads to sharper segmentation at the expense of a small amount of added computation.

4.3. Motion and Stereo Disparity

For motion segmentation (or binocular segmentation for a stereo pair), the nodes of the graph are the triplet (x, y, t), where (x, y) denote image location and t is time. The weights between two nodes describe the similarity of the motion at the two pixel locations at that time. We propose to compute these weights softly through *motion profile*. Instead of trying to determine exactly where each pixel moves to in the next frame (as in optical flow), we compute a *probability distribution* over the locations where the pixel might move to. Similarity between two nodes is then measured as the similarity of the motion profiles.

This technique can be made computationally efficient for long image sequences by considering only a fixed number of image frames centered around each incoming image frame in the time domain to compute the segmentation. Because there is a significant overlap of the image frames used to compute the segmentation from one time step to another, we can use it to our advantage to speed up our computation. Specifically, when solving the generalized eigensystem using the Lanczos method, the eigenvectors from a previous time step can provide us with a good guess for the initial vectors at the next time step, and we can arrive at the solution very quickly. An example of the motion segmentation results for the flower garden sequence is shown in 4. For details, please refer to [11].

5. RESULTS AND DISCUSSION

Results are shown in Figure 2 using texture and intensity and in Figure 3 using contour continuity. For more results, the reader is encouraged to look at our Web site: *http://www.cs.berkeley.edu/~jshi/Grouping/*.

In this paper, we have proposed a coherent system for image and video segmentation based on the normalized cut framework. Since finding the best partition is equivalent to computing eigenvectors, the algorithm is efficient. Moreover, there are many methods for multiresolution implementation of the segmentation. Due to space limitations, details are omitted here.

A good image/video segmentation system has numerous applications. It has not escaped our notice that the segmentation system we have described immediately suggests an efficient method of image and video compression. Another major application is image retrieval in large image databases such as in [1].

6. ACKNOWLEDGEMENTS

This work was supported by an NSF Digital Library Grant (IRI 94-11334), (ARO) DAAH04-96-1-0341, an NSF Graduate Fellowship for J.S. and S.B., and a U.C. Berkeley Chancellor's Opportunity Predoctoral Fellowship for S.B.

7. REFERENCES

[1] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using the expectation-maximization algorithm and its application to content-based image retrieval. In *Proc. Int. Conf. Computer Vision*, Bombay, India, Jan. 1998.

[2] S. Belongie and J. Malik. Finding boundaries in natural images: a new method using point descriptors and area completion. In *submitted to ECCV98*, 1998.

[3] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8, 1986.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Statistical Society*, 39(B), 1977.

[5] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Optical Society of America*, 7(2):923–32, May 1990.

[6] M.C. Morrone and R.A. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–13, 1987.

[7] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions, and associated variational problems. *Comm. Pure Math.*, pages 577–684, 1989.

[8] P. Parent and S.W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(8):823–39, Aug. 1989.

[9] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proc. Int. Conf. Computer Vision*, pages 52–7, Osaka, Japan, Dec 1990.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–7, San Juan, Puerto Rico, June 1997.

[11] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proc. Int. Conf. Computer Vision*, Bombay, India, Jan. 1998.

[12] S. Ullman. Filling-in the gaps: the shape of subjective contours and a model for their generation. *Biological Cybernetics*, 25:1–6, 1976.

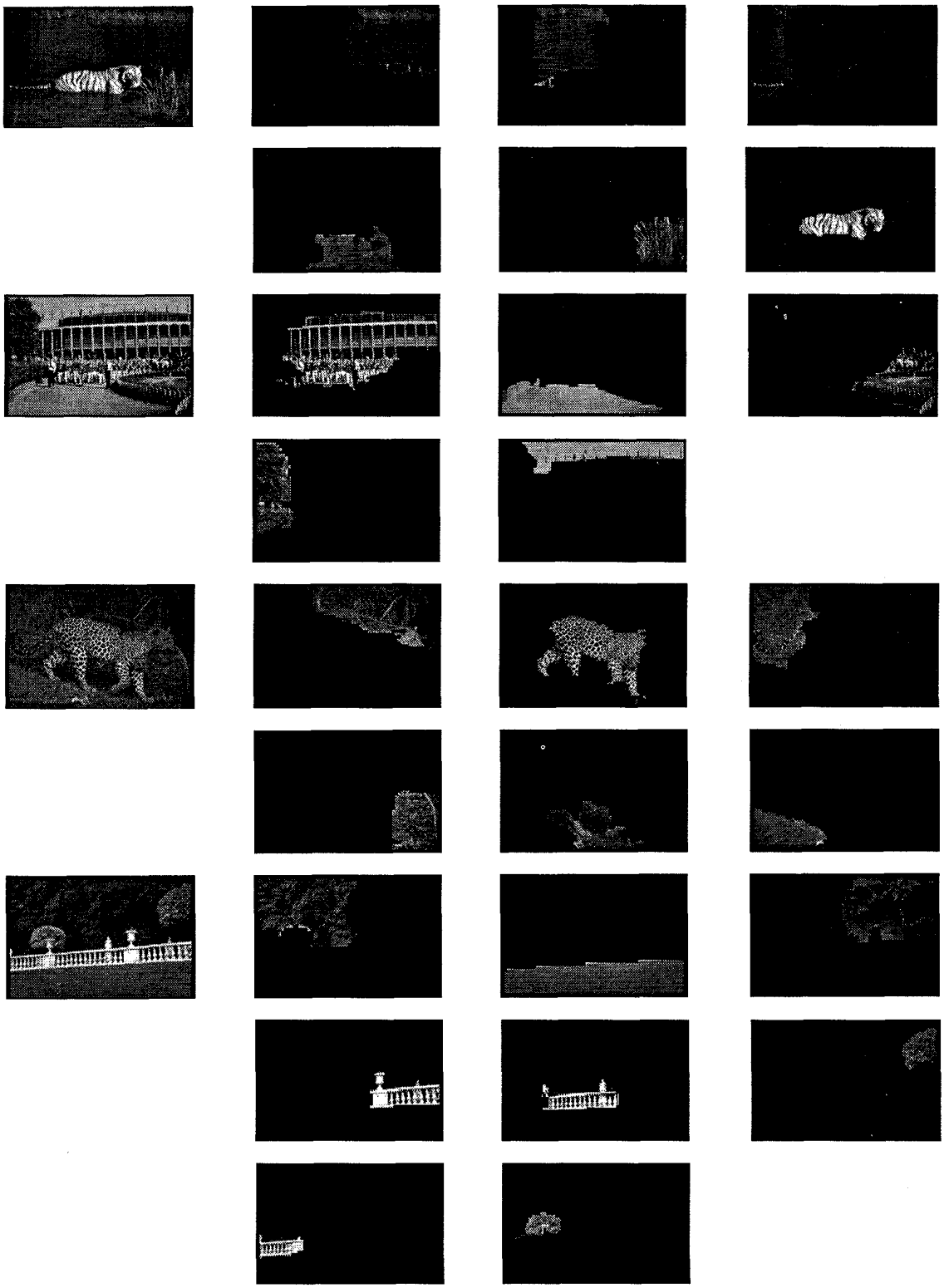

Figure 2: Segmentation using intensity and texture. Original image shown on the left and the segments on the right.

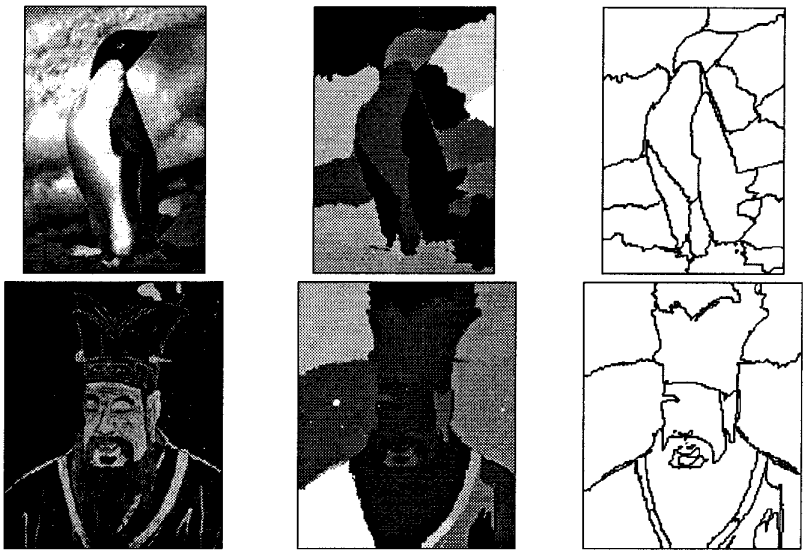

Figure 3: Segmentation based on intensity and contour continuity. Left: original image; middle: segments; right: boundaries of segments.

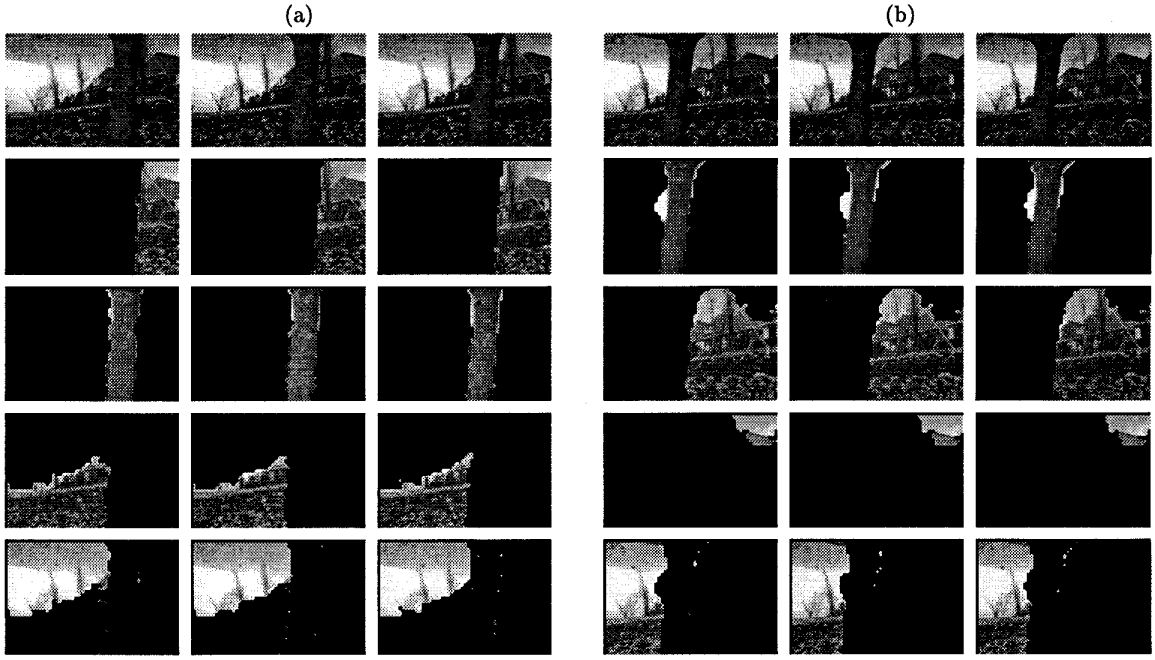

Figure 4: Subplot (a) shows three of the first six frames of the "flower garden" sequence along with the segmentation. The original image size is 120×175, and image patches of size 3×3 are used to construct the partition graph. Each of the image patches are connected to others that are less than 5 superpixels and 3 image frames away. Subplot (b) shows the 15th to the 18th frame of the sequence and the motion segmentation using tracking algorithm with the sliding time window method.