SHAPE TRAINING FOR VIDEO OBJECT SEGMENTATION

Daehee Kim and Yo-Sung Ho

Kwangju Institute of Science and Technology
1 Oryong-dong Puk-gu, Kwangju, 500-712, Korea
kimdh@.kjist.ac.kr, hoyo@kjist.ac.kr

ABSTRACT

Since most algorithms for automatic video segmentation cannot extract video objects in a picture frame accurately, we can take a user-assisted approach in generating VOPs of moving objects. In this paper, we propose a semi-automatic video segmentation algorithm using semantic information. In order to reduce effects of unwanted feature points due to the low-level image processing operations, we employ an active contour with B-spline curves. In addition, we define an external energy with the SUSAN feature detector whose computational complexity is lower than popular morphological filtering operation.

1. INTRODUCTION

The MPEG-4 visual standard [1] enables content-based functionalities by introducing the concept of the video object plane (VOP). A VOP is defined as a coding unit of the MPEG-4 natural visual coding, determined by the shape of the predefined video object, at certain frame over an entire sequence. In order to process the image based on its contents, we should partition the image into a set of meaningful objects and determine their parameters, such as color, shape and motion because the MPEG-4 assumes that VOPs are available prior to the encoding process. For content-based video representation, each frame of the input sequence is segmented into semantically meaningful objects or video contents of interest. Knowledge about the shape of each video object in the scene can help us to reconstruct subjectively better images.

Among the conventional approaches for video segmentation, the spatio-temporal segmentation technique is particularly interesting to us because the MPEG-4 visual standard includes a typical spatio-temporal algorithm in its informative annex [1]. This algorithm extracts edge information by the morphological gradient operation and obtains information of moving objects by the change detection mask.

An MPEG-4 visual encoder requires individual video objects that are discriminated by each binary mask corresponding each individual video object. However, the spatio-temporal segmentation algorithm does not allow us to extract accurate boundaries of individual video objects from multiple objects in a single frame. Since it is not easy to define a mathematical model and a similarity measure for extracting video objects adequately, this automatic segmentation algorithm cannot provide satisfactory segmentation results from different kinds of image sequences. If the user can define VOPs in the first frame in a user-assisted manner, we may obtain better segmentation results in the subsequent picture frames.

Therefore, the user-assisted segmentation approach is more practical in generating VOPs of moving objects. An active contour method is one of the user-assisted segmentation approaches that allows user interactivity to obtain the semantic information about arbitrary video objects. In this paper, we propose a new video segmentation algorithm based on the active contour algorithm to find the shape of visual objects and to track them accurately.

2. THE B-SNAKE ALGOITHM

2.1. B-Spline Function

In the active contour or snake algorithm, we try to find an energy-minimizing curve from an initial curve indicated by the user. Performance of the active contour algorithm depends mainly on the definition of the energy functional $E(s)$. The shape of the active contour is controlled by internal, external and constraint forces. While the internal force is the smoothness constraint on the curve, the external force guides the active contour towards image features. The constraint force allows interactivity in manipulating the active contour. The energy functional $E(s)$ is represented as a parametric curve $\mathbf{r}(s)=(x(s),y(s))$, where s is the parameter for the given interval [2].

A functional of the snake is defined by

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{r}(s))ds$$
$$= \int_0^1 [E_{int}(\mathbf{r}(s)) + E_{ext}(\mathbf{r}(s)) + E_{con}(\mathbf{r}(s))]ds \quad (1)$$

where E_{int}, E_{ext} and E_{con} represent the internal energy of the contour, the external image force and the constraint force, respectively. The final location of the active contour corresponds to the local minimum of the energy functional.

Since $E(s)$ is computed over the discrete grid space, we need to discretize the continuous curve. Therefore, for the active contour algorithm, we can define the internal energy function by

$$E_{int} = \sum_i \alpha_i |\mathbf{r}_i - \mathbf{r}_{i-1}|^2 + \beta_i |\mathbf{r}_{i+1} - 2\mathbf{r}_i + \mathbf{r}_{i-1}|^2 \quad (2)$$

If there is no external energy in this approximation, the contour could shrink into a single point. In order to prevent this situation, we can use a variant for the internal energy function. Since a finite difference approximation conveys insufficient information on the shape of the curve between samples, modern numerical analysis techniques employ the finite element method, where variables $\mathbf{r}(s_i)$ are regarded as control parameters from which the continuous curve $\mathbf{r}(s)$ can be reconstructed completely.

We can obtain a smooth approximation of the curve by modeling $\mathbf{r}(s)$ as a polynomial spline curve that passes near but not necessarily through the control points. This is partially efficient because the spline maintains a degree of smoothness which is related to the internal energy function. However, the spline curve can fail to adjust sharp corner points because external image features cannot be reflected into the spline curve. A practical solution for this problem is the B-spline curve, which can release the smoothness condition at certain points on the curve with multiple knots and multiple control points [3].

The parametric curve $\mathbf{r}(s)=(x(s), y(s))$ is a particular function of a parameter s. A spline basis function of order d is defined as a piecewise continuous polynomial function, consisting of d concatenated polynomial segments that are joined together at breakpoints. A simple shape can be approximated by a polynomial curve with a few segments. More complex shapes can be accommodated by high-order polynomials. In general, the polynomial order is fixed to quadratic (d=3) or cubic (d=4).

A B-spline function $x(s)$ can be constructed as a weighted sum of N_B basis functions $B_n(s)$, where n=0,...,N_B-1. By setting d=3, the curve has a continuous gradient. Therefore, the constructed spline function satisfies the internal energy requirement naturally.

The spline function can be represented by

$$x(s) = \mathbf{B}(s)^T \mathbf{Q}_x \quad (3)$$

$$\mathbf{B}(s) = [B_0(s), B_1(s), \ldots, B_{N_B-1}(s)]^T$$
$$\mathbf{Q}_x = [x_0, x_1, \ldots, x_{N_B-1}]^T$$

where x_n is the weight for a basis function $B_n(s)$. Therefore, the parametric curve $\mathbf{r}(s)$ is also represented in the matrix following form.

$$\mathbf{r}(s) = \mathbf{U}(s)\mathbf{Q} \quad (4)$$

where

$$\mathbf{U}(s) = \mathbf{I}_2 \otimes \mathbf{B}(s)^T = \begin{pmatrix} \mathbf{B}(s)^T & 0 \\ 0 & \mathbf{B}(s)^T \end{pmatrix} \quad (5)$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_x \\ \mathbf{Q}_y \end{pmatrix} \quad (6)$$

The control vector $\mathbf{Q}$ is constructed from x and y components of all control points.

2.2. Shape Space Model

When we try to fit the initial curve to the desired one, it is not efficient to describe the variation of the curve by the curve itself or control points because the curve itself has an infinite number of points and the vector of control points has the dimension N_Q=$2N_B$.

A shape space of dimension N_X, which is considerably smaller than N_Q, is appropriate to describe variations of the curve. The shape space is defined as a set of allowed deformation from the base curve. The shape space is constructed from a set of vectors of dimension N_X. It is desirable to restrict the displacement of control points to a lower dimensional shape space if it preserves the frame of the shape. An unconstrained control vector $\mathbf{Q}$ may lead to unstable active contours [4].

A change of the curve in the shape space is a linear mapping of the shape space vector $\mathbf{X}$ to a control vector $\mathbf{Q}$

$$\mathbf{Q} = \mathbf{W}\mathbf{X} + \mathbf{Q}_0 \quad (7)$$

where $\mathbf{W}$ is a $N_Q \times N_X$ shape matrix, and $\mathbf{Q}_0$ is a control vector of the initial curve. A shape space vector $\mathbf{X}$ describes the change of the initial curve.

The generalized linear mapping for the active contour is computationally simple. This approach works well for rigid objects or simple non-rigid objects. Linearity can certainly be a limitation when the motion of an object becomes more complex. However, moving objects over successive frames do not change significantly even if objects are non-rigid. Therefore, curve fitting and tracking procedures can be simplified as linear operations.

In this paper, we describe contour changes by the 6-parameter affine model ($N_X = 6$). The affine model can be viewed as a class of linear transformations that can be applied on the initial curve $\mathbf{r}_0(s)$:

$$\mathbf{r}(s) = \mathbf{u} + \mathbf{M}\mathbf{r}_0(s) \tag{8}$$

where $\mathbf{u}=(u_1, u_2)^T$ is a two-dimensional translation vector, and $\mathbf{M}$ is a 2×2 matrix. Therefore, $\mathbf{M}$ and $\mathbf{u}$ have six degrees of freedom to describe the transformation of the initial curve $\mathbf{r}_0(s)$.

This class can be represented by a shape space with the initial curve $\mathbf{Q}_0$ and the shape matrix:

$$\mathbf{W} = \begin{pmatrix} 1 & 0 & Q_{x0} & 0 & 0 & Q_{y0} \\ 0 & 1 & 0 & Q_{y0} & Q_{x0} & 0 \end{pmatrix} \tag{9}$$

The first two columns of $\mathbf{W}$ represent horizontal and vertical translation. Each column of $\mathbf{W}$ forms a basis vector of the shape space, but it is not necessary to be orthogonal to the other vectors.

2.3. Minimization Process

In order to find boundaries of video objects, we need to define a distortion measure that can be expressed by the curve norm of shape differences between the estimated curve $\mathbf{r}(s)$ and the desired curve $\mathbf{r}_d(s)$.

$$\left\|\mathbf{r}(f(s)) - \mathbf{r}_d(s)\right\|^2 = \frac{1}{L}\int_0^L \left|\mathbf{r}(f(s)) - \mathbf{r}_d(s)\right|^2 ds \tag{10}$$

where L is the length of the interval of s. Therefore, our goal is to fit the estimated curve to the desired curve in such a way that the vector difference function would ideally have a norm of zero.

In Eq. (10), $f(s)$ is the adjustment function of the parameter s. It is necessary for the function $f(s)$ to be matched to the starting points of two curves and to make the traveling speed along one curve be the same as the traveling speed of the other curve.

Mathematically, the optimal function $f_{opt}(s)$ can be obtained by Eq. (11), whose computation is feasible if we search for a local minimum rather than the global minimum.

$$f_{opt}(s) = \arg\min_f \frac{1}{L}\int_0^L \left|\mathbf{r}(f(s)) - \mathbf{r}_d(s)\right|^2 ds \tag{11}$$

If two curves are very similar, we can approximate $\mathbf{n}(f(s))$ by $\mathbf{n}(s)$, where $\mathbf{n}(f(s))$ is the normal vector to the tangent vector of $\mathbf{r}(s)$. Therefore, $\|\mathbf{r}(s) - \mathbf{r}_d(s)\|$ can be replaced by $[\mathbf{r}(s) - \mathbf{r}_d(s)]\cdot\mathbf{n}(s)$ [4]. If the integral of Eq. (10) is approximated by a summation and the normal vector $\mathbf{n}(s)$ is used to measure the shape difference, our minimization criterion can be changed to

$$\left\|\mathbf{r} - \mathbf{r}_d\right\|^2 \approx \frac{1}{N}\sum_{i=1}^{N}\left[(\mathbf{r}_d(s_i) - \mathbf{r}(s_i))\cdot\mathbf{n}(s_i)\right]^2 \tag{12}$$

where N is the number of regularly spaced points on the interval of s. Eq. (12) can be rewritten in terms of the initial estimated curve $\mathbf{r}_0$ and the shape space vector.

$$\left\|\mathbf{r} - \mathbf{r}_d\right\|^2 \approx \frac{1}{N}\sum_{i=1}^{N}[(\mathbf{r}_d(s_i) - \mathbf{r}_0(s_i))\cdot\mathbf{n}(s_i) - \mathbf{n}^T\mathbf{U}(s_i)\mathbf{W}(\mathbf{X} - \mathbf{X}_0)]^2 \tag{13}$$

In order to minimize $\|\mathbf{r}(s) - \mathbf{r}_d(s)\|^2$ in Eq. (13) and find the estimated shape space vector $\mathbf{X}^*$, we employ the least square approach. The minimal $\mathbf{X}^*$ can be estimated by setting $\partial\|\mathbf{r}-\mathbf{r}_d\|^2/\partial\mathbf{X} = 0$, which leads to

$$\mathbf{X}^* = \left(\sum_{i=1}^{N}\rho_i\mathbf{W}^T\mathbf{U}^T\mathbf{n}\mathbf{n}^T\mathbf{U}\mathbf{W}\right)^{-1}\left(\sum_{j=1}^{N}\rho_i\mathbf{W}\mathbf{U}^T\mathbf{A}^T\mathbf{n}(\mathbf{r}_d - \mathbf{r}_0)^T\mathbf{n}\right) \tag{14}$$

Once we find $\mathbf{X}^*$, we can obtain the optimal control vector $\mathbf{Q}$ by Eq. (7). For more accurate results, we can repeat the same procedure several times by setting the previously fitted curve $\mathbf{r}(s)$ as the initial curve $\mathbf{r}_0(s)$.

3. FEATURE EXTRACTION

Conventional active contour algorithms are designed for extracting objects in the homogeneous background; however, they may not work well for objects with the complex background. In this paper, we propose new image features as the energy function for images of complex background.

If we use a simple edge detector that most conventional active contour algorithms employ, it is difficult to obtain satisfactory segmentation results from images of complex background. In this paper, we employ the smallest univalue segment assimilating nucleus (SUSAN) operator to extract image features. We also define the external energy function as the extracted image features. Although the SUSAN edge detector requires lower computational complexity than morphological gradient tools or the Canny edge detector, it works well for images of complex background.

The SUSAN edge detector consists of three steps [5]. In the first step, we place the nucleus of a circular mask

around a pixel that is being tested. In the second step, we calculate the number of pixels within the circular mask that have similar brightness to the nucleus by

$$c(\mathbf{p},\mathbf{p}_0)=\exp(-\frac{I(\mathbf{p})-I(\mathbf{p}_0)}{t})^6 \qquad (15)$$

where $\mathbf{p}$ is the position of an arbitrary pixel within the circular mask and $\mathbf{p}_0$ is the position of the nucleus. The number of pixels which have similar brightness to the nucleus is defined as the area of segment assimilating nucleus (USAN). In order to reduce computation time of the exponential function, we make a lookup table for the possible values of $I(\mathbf{p}) - I(\mathbf{p}_0)$.

$$n(\mathbf{p}_0)=\sum_{\mathbf{p}} c(\mathbf{p},\mathbf{p}_0) \qquad (16)$$

In the third step, we subtract the size of USAN from the geometric threshold to produce an image of edge strength by

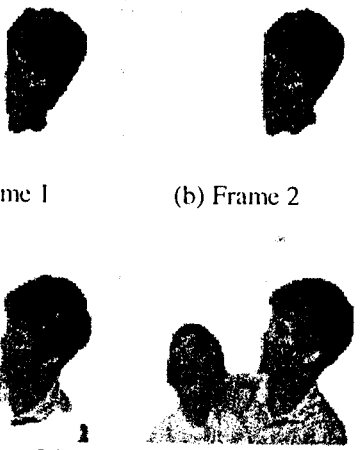
$$R(\mathbf{p}_0)=\begin{cases} g-n(\mathbf{p}_0) & \text{if } \mathbf{p}_0 < g \\ 0 & \text{otherwise} \end{cases} \qquad (17)$$

where $R(\mathbf{p}_0)$ is the initial edge response and g is a fixed geometric threshold. When we find edges in the absence of noise, we do not need the geometric threshold. However, g is set to $3n_{max}/4$ for optimal noise rejection [6]. After SUSAN edge detection, we utilize the edge image to extract the imaginary feature curve $\mathbf{r}_d(s)$.

4. EXPERIMENTAL RESULTS

In order to evaluate our proposed algorithm, we perform computer simulations on MOTHER AND DAUGHTER of the CIF format. Fig. 1 shows simulation results of the spatio-temporal segmentation algorithm that is included in the Annex F of the MPEG-4 visual standard [1]. The automatic segmentation algorithm has transient responses in the first 30 to 50 image frames. Since the initial period is spent to find video objects, segmentation results during the transient time period are not meaningful.

In addition, an MPEG-4 visual encoder requires individual video objects that are discriminated by each binary mask corresponding each individual video object. While an MEPG-4 visual decoder decodes each video object and composites the entire frame using all decoded video objects, an MEPG-4 visual encoder encodes each video object with each corresponding binary mask independently. However, it is difficult for spatio-temporal segmentation algorithms to separate individual video objects from multiple objects and identify individual video objects.

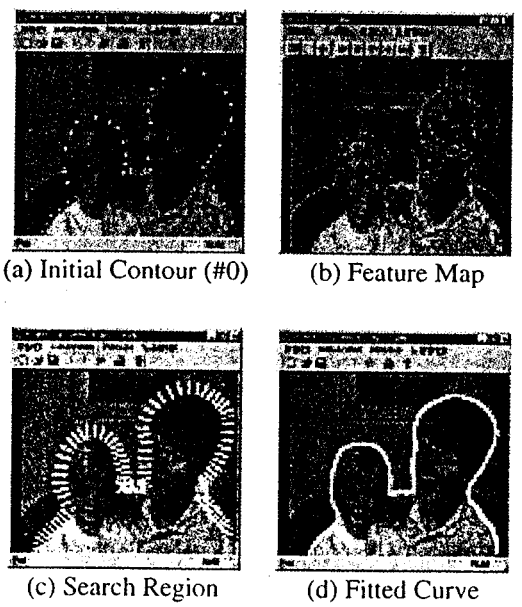


(a) Frame 1 (b) Frame 2

(c) Frame 34 (d) Frame 51

Fig. 1 Automatic Segmentation Results

Fig. 2(a) shows the initial contour provided by the user-pointing device, such as a mouse. Image features obtained by the SUSAN edge detector are displayed in Fig. 2(b), where we observe simplified feature maps of the image. The search region displayed in Fig. 2(c) is formed by sweeping normal vectors along the initial curve. In Fig. 2(c), points on white lines are candidates of $\mathbf{r}_d(s_i)$ corresponding to $\mathbf{r}(s_i)$. Fig. 2(d) is the final segmentation result.



(a) Initial Contour (#0) (b) Feature Map

(c) Search Region (d) Fitted Curve

Fig. 2 User-Assisted Segmentation Results

The user-assisted algorithm does not generate transient responses. Control points in the second frame are

estimated from those in the first frame, as shown in Fig. 2(d).

Since most automatic segmentation algorithms are based on frame differences, objects corresponding to the mother and the daughter are not separated by those automatic segmentation algorithms, as shown in Fig. 1. However, in the user-assisted segmentation algorithm, if we want to extract several individual objects from multiple objects in a single frame, we just draw initial contours around each interesting video object and apply the same active contour algorithm to each video object.

Fig. 3 demonstrates the segmented daughter by the proposed semi-automatic segmentation algorithm. If we want to extract the daughter only from the image, we draw an initial contour around the daughter and apply the active contour algorithm. Therefore, we can obtain the binary mask corresponding the daughter from the result of Fig. 3 and this mask can be employed as the input of a MPEG-4 visual encoder to identify an individual video object.

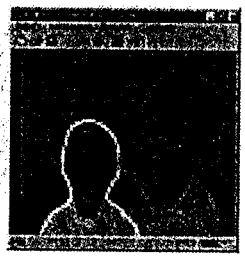


Fig 3. Object Extraction from Multiple Objects

Fig. 4 shows segmentation results for NEWS sequence. After interesting video objects are recognized in the first frame, the control points of the fitted curve are transferred to the next frame. We estimate control points of the next frame using the control points of the fitted curve at the first frame or the previous frame. We set these points as control points of the initial contour in the next frame, and obtain the video object by the proposed segmentation algorithm. This procedure is performed repeatedly until the end of the image sequence. The results are displayed every 18 frames in Fig. 4.

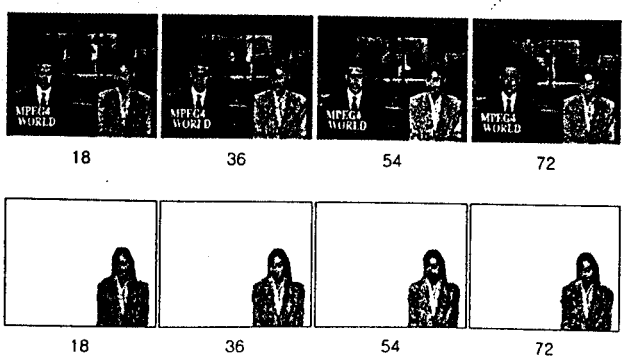


Fig 4. Segmentation Results for NEWS

5. CONCLUSIONS

In this paper, we have proposed a new user-assisted active contour algorithm to extract video objects from image sequences. Since we employ the SUSAN edge detector, the proposed algorithm can be applied to images of the complex background. In addition, owing to the shape space vector for describing the change of curves, we can ignore some outliers or unwanted feature points generated by low-level image processing operators. The proposed user-assisted active contour algorithm has no transition responses and can extract individual video objects over multiple objects in the same frame independently. We employ the active contour algorithm based on the B-spline basis function to extract accurate object boundaries efficiently.

6. ACKNOWLEDGMENTS

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks (UFON) Research Center at Kwangju Institute of Science and Technology (K-JIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

7. REFERENCES

[1] ISO/IEC FDIS 14496-2: "Information technology - generic coding of audio-visual objects, Part 2: visual," *ISO/IEC JTC1/SC29/WG11*, Oct. 1998.

[2] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models," *First International Conference on Computer Vision*, pp. 259-269, 1987.

[3] J. D. Foley, A. Dam, S. K. Feiner, J. F. Hughes and R. L. Phillips, *Introduction to Computer Graphics*, Addison-Wesley, New York, 1995.

[4] A. Blake and M. Isard, *Active Contours*, Springer, London, 1998.

[5] S.M. Smith, "Flexible filter neighborhood designation," *Proc. 13th Int. Conf. on Pattern Recognition*, vol.1, pp. 206-212, 1996.

[6] http://www.fmrib.ox.ac.uk/fsl/susan