© *The Visual Computer, vol. 12, no 2, 1996, pp 75-83.*

A practical solution for tracking edges in image sequences with snakes

Michael Hoch* , Peter C. Litwinowicz

Apple Computer Inc.
ATG-Graphics
1 Infinite Loop, MS 301-3J
Cupertino, California 95014

Abstract - Active contour models, or "snakes," developed in (Kass et al. 1988), use a simple physical model to track edges in image sequences. Snakes as originally defined however, tend to shrink, stretch and slide back and forth in unwanted ways along a tracked edge and are also confused by multiple edges, always grabbing the nearest one. In this paper a practical solution is presented that combines motion estimation techniques with snakes to overcome these problems. An algorithm is presented that uses a block matching technique to guide the endpoints of the snake, optical flow to push the snake in the direction of the underlying motion, followed by the traditional snake edge-fitting minimization process. We use this technique for tracking facial features of an actor for driving computer animated characters.

Key words: Pattern Recognition - Edge Tracking - Active Contour Models - Facial Animation

* Academy of Media Arts, Deptartment of Computer Science/Audio-Visual Media, Peter-Welter-Platz 2, 50676 Cologne, Germany, e-mail: micha@khm.uni-koeln.de

Correspondence to: M. Hoch

1 Introduction

Feature tracking in video sequences is of proven use for robot vision, medical applications and animation. The tracking algorithms often involve finding contours because in many cases contours provide important information about an object or a scene. For tracking contours in multiple images, active contour models (snakes) have been successfully used in various applications (Kass et al. 1988 and Blake and Yuille 1992). However, depending on the application several problems have to be solved to create a practical tracking system based on snakes. In this paper we will focus on the problem of tracking edges in video sequences corresponding to facial features of an actor. The tracked facial expressions may then be used for driving computer animated characters (DeWitt and Edelstein 1982, Williams 1990, Patterson et al. 1991, Terzopoulos and Waters 1991). Using snakes for tracking facial features like lips and eyebrows was proposed in (Kass et al. 1988); a more elaborate tracking driving a 3D face model is described in (Terzopoulos and Waters 1991). In our application we wish to track the facial expressions of an actor so that we may apply the motion to images of 2D characters (the mapping technique is described in Patterson et al. 1991, and the image warping technique is described in Litwinowicz and Williams 1994).

We will first explain the problems that need to be solved for our animation application. We will then present a short description of other tracking algorithms and how they can be used in combination to enhance the tracking performance of snakes, solving some key problems for tracking edge features in our application. Finally we will present some animation results and show that the presented technique is useful for edge tracking in arbitrary video sequences as well.

2 Discussion of snakes and their problems for our application

A snake is an "energy-minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges" (Kass et al. 1988). The initial feature edges are outlined by the user in the first frame of a sequence. The algorithm, as originally published, tracks features by using forces derived from the gradient of the image (termed "image forces"). These forces push a snake towards the nearest edge in an image. The

edge finding process is repeated for each frame of a sequence to track the features delineated in the first frame.

During this process the endpoints of the snakes may move away from the corresponding endpoint features in the first frame, that is, the endpoints of the snakes can slide back and forth along an edge defined by a feature (Fig. 1a). In figure 1a three frames of a person talking and a snake tracking the upper edge of the lower lip are shown. It should be noted that we use NTSC video, thus our input video rate is 30 frames/sec., and that all figures in the paper use successive frames (that is, no frames were skipped). The actor's expressions were painted with high contrast makeup to facilitate tracking. Snakes are very good at finding movement of an edge in a normal direction, but poor at tracking tangential movement. For our application it is important that the snake endpoints track the corresponding feature endpoints in the image sequence. If our animation algorithms are presented with “sloppily” tracked motion (the endpoints slide back and forth along their image features) the resulting animations will not be very convincing because regions of the character’s face will slide along with them.

If features in the video sequence move far enough from one frame to another, a snake may actually switch edges. For instance, in viewing video of a person talking it is not uncommon to see the lower edge of the upper lip visually replace the upper edge of the lower lip in succeeding frames (Fig. 1b). Without motion prediction and compensation of some sort, a snake trying to track the upper lip will suddenly find itself tracking the lower lip (Fig. 1b). Because of these problems, extensive user interaction may be necessary to extract motion from a video sequence.

If features squash and stretch, as they do on our actor's face, snakes that have a length preserving constraint are of little use. Using active contours that incorporate relaxed, or rest configurations for the contours (Blake and Yuille 1992) will overcome the problem that an unconstrained snake relaxes back to a straight line (or even shrinks to a point), but requires us to build or infer a more complex model before we start the tracking process. Other extensions or variations of snakes deal with the physically motivated formulation of snakes or reduce the computational costs of snakes or propose a probabilistic

interpretation for extracting reliable information from noisy observations (Blake and Yuille 1992), but do not solve the moving endpoint problem. Most examples of non-closed snakes clearly show the limitations of unenhanced snakes in adhering to the endpoints of a tracked edge (see examples in Blake and Yuille 1992 and Terzopoulos and Waters 1991).

We introduce the use of block matching techniques to track and position the endpoints of snakes. After block matching, the endpoints of a snake are held in place while the energy minimization process takes place. This combination of techniques overcomes the problem of snakes sliding back and forth along their corresponding texture features. To overcome the problem of the snake finding an incorrect edge due to large motion, we apply for the first time the use of optical flow techniques in conjunction with snakes. Optical flow techniques can use all of the local texture and shading information to estimate the motion in a region, but this estimate can be refined for the high contrast feature of a tracked edge. After the optical flow estimate is used to push a snake near to its target edge, the snake minimization algorithm is employed to conform to the edge more accurately. In the following sections we will first briefly describe how other motion estimation techniques are used in tandem with the snake algorithm, i.e. block matching and optical flow. Then we will outline the semi-automatic system and present some results. A discussion of future work and a conclusion will be presented at the end of the paper.

3 Motion estimation techniques

Block matching:

The first step in the tracking process is to use block matching to find the new locations for the endpoints of a snake, starting from their positions in the previous frame. Block matching is commonly used for motion analysis to find correspondence among local image patterns in a sequence of images. Applications include template matching, stereo correspondence and motion compensation for video compression purposes (Lim 1990, Niemann 1990). Pixel-by-pixel block matching can be used to estimate optical flow, but is computationally expensive without dedicated hardware. We apply this accurate, but expensive, estimator very sparsely, at the endpoints of each snake. Less expensive, less accurate estimators can be used to fit the body of the snake once endpoints have been optimally matched. In our application we use a 13x13

block and a search area of 9x9 pixels (block size and the size of the search area were found by experimentation and worked well for our video sequences). In order to find the best match between blocks withing the search area a measure of simularity between patterns is needed:

$$C(i,\delta) = \sum_m W(m)F[f(i+m)] \bullet F[h(i+m+\delta)] \tag{1}$$

Where correlation C is computed between a pattern in the reference image f centered at point i and a pattern in the current image h centered at point $i+\delta$. The size of the pattern is determined by a window function W (Burt et al. 1982). A preprocessing operator F is applied to the current and the reference frame; the comparison operator "$\bullet$" can be any operator that computes pixel to pixel comparison, e.g. subtraction, multiplication, or more elaborate operators. Among the known measurements of similarity are absolute differences and squared differences which tend to identify only identical image patterns and are highly sensitive to changes in illumination and reflectance. To get the best performance on the correspondence itself, we choose variance-normalized correlation with a Gaussian-window (Burt et al. 1982):

$$C_v(i,\delta) = \frac{\sum W(m)\left(f(i+m) - \bar{f}(i)\right)\left(h(i+m+\delta) - \bar{h}(i+\delta)\right)}{\sqrt{Var_i(i)}\sqrt{Var_h(i+\delta)}} \tag{2}$$

Where $\bar{f}$ and $\bar{h}$ are the mean values of the block beeing considered, *Var* denotes the respective variance, and simple multiplication is used as the comparison operator. Furthermore we assume that no features disappear, that a variational measure of brightness is conserved for each feature (be it pixel, edge or region), and that the motion is assumed to be smooth in the time domain (Li et al. 1993). In order to achieve higher accuracy without compromising efficiency, we first determine the best match for integer (pixel) displacements. We then refine that initial estimate to subpixel resolution by evaluating a smaller block of 5x5 pixels. Figure 1c shows the same sequence as Fig. 1a but with the addition of block matching on the snake endpoints. This is a good, but in our experience typical, example of the way in which block matching augments the snake's tangential tracking performance.

Gradient-based Optical Flow:

The next step in the tracking process is to automatically push the interior control points (that is, non-endpoints) of a snake to wherever their corresponding image edge has moved, using optical flow techniques. Optical flow is based on the assumptions that illumination is constant and that occlusion can be ignored, that is, the observed graylevel changes are only due to the motion of the underlying objects. In this case it is evident that

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t) \tag{3}$$

Where I is a greyscale image sequence, parameterized by pixel locations (x, y) and time t. Expanding the right side of (3) into a Taylor series and neglecting the higher order terms yields the so called motion contraint equation (Niemann 1990).

$$f_x v_x + f_y v_y + f_t = 0 \tag{4}$$

where f_x and f_y are the directional gradients and v_x and v_y denote the velocities in x and y respectively. Since there are two velocity components but only one equation, equation (4) is not sufficient to compute the velocities. In optical flow techniques an additional constraint is introduced: a smoothness constraint assumes that the flow field is smooth in a certain local area. The optical flow method used is described in (Bergen and Hingorani 1990), which uses a pyramid of successively low-passed versions of the gradient to compute the optical flow. To find the motion $v = (v_x, v_y)$ at a point (x, y) the following error term is minimized:

$$E = \sum \left(f_x v_x + f_y v_y + f_t \right)^2 \tag{4}$$

Where summation is over a 5 x 5 region centered about the point (x, y). Differentiating and setting the derivatives to zero yields the equation

$$\begin{bmatrix} \sum F_x^2 & \sum F_x F_y \\ \sum F_x F_y & \sum F_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} -\sum F_x F_t \\ -\sum F_y F_t \end{bmatrix} \tag{5}$$

$$W \qquad v = \gamma$$

This equation can be solved for v_x and v_y using an eigenvector approach. The sliding summation of the region can efficiently implemented by using only

4*wh* additions for an image of size *w x h*. Motion vectors are first estimated using the low frequency portions of the gradient pyramid (accounting for large, coarse motion), and refined by using the levels of the pyramid with higher frequency content. We do not go into detail concerning the optical flow technique because any optical flow method can be used at this point. For details on optical flow the reader is referred to (Bergen and Hingorani 1990).

Calculating optical flow for two successive images of the sequence results in motion vectors for each pixel. Hence, for each control point along the snake we can determine where the underlying pixel has moved and push the snake points in the appropriate direction. We used an optical flow method because, although it is less accurate than block-matching at every pixel, it is independent of the number of snakes (and the total number of snake control points), and requires less compute time. (We considered block matching for each control point along a snake, but the accuracy was unnecessary and was much more time consuming). Figure 1d shows the same sequence as Fig. 1a but with the addition of the optical flow technique. The snake, after applying the motion vectors in the optical flow field, will usually be near enough to the appropriate edge for the snake energy optimizing process to find the correct edge.

4 Snake energy minimization process

After nailing the endpoints down and pushing the interior points towards the appropriate edge, the next and last automatic step in the tracking process allows each snake to find its rest position. Note that the block matching and optical flow techniques are used only as preprocessing steps to the snake algorithm, and are not used *during* the snake energy minimization process (the block matched endpoints are, however, held in place during the minimization process).

To effectively use snakes to track edges in an image one must preprocess the image. Usually a Sobel or other edge enhancement filter is run over the image and the gradient of the enhanced image is calculated to determine the direction of the nearest edge. A description of edge enhancement filters can be found in (Pratt 1991). If the nearest edge is "far" away the gradient may be zero and the snake will not gravitate towards it. A smoothing filter such as a gaussian convolution may be applied. As described in (Kass et al. 1988) the snake

relaxation process minimizes energy using the snake's internal stiffness parameters while at the same time using the external image forces as supplied by the smoothed gradient. In doing this each snake is guided toward its nearest edge while at the same time maintaining its appropriate stiffness. Each point along a snake is pushed at most one pixel at a time by the gradient image, "animating" them in position, so as to allow the user to guide them to another edge (with spring forces attached to the mouse) if so desired.

5 Semi-automatic system

In Fig. 2 a block diagram of the semi-automatic system is shown. The 2½D animation system Inkwell (Litwinowicz 1991) is used to outline the initial feature curves on the first frame of the video sequence; this defines the initial positions of the snakes. Next, the tracking method is applied which incorporates the block matching procedure to track the endpoints of the snakes, optical flow to push the snake in the right direction, and the snake relaxation process to allow the snake to find its nearest (and at this point, hopefully correct) edge.

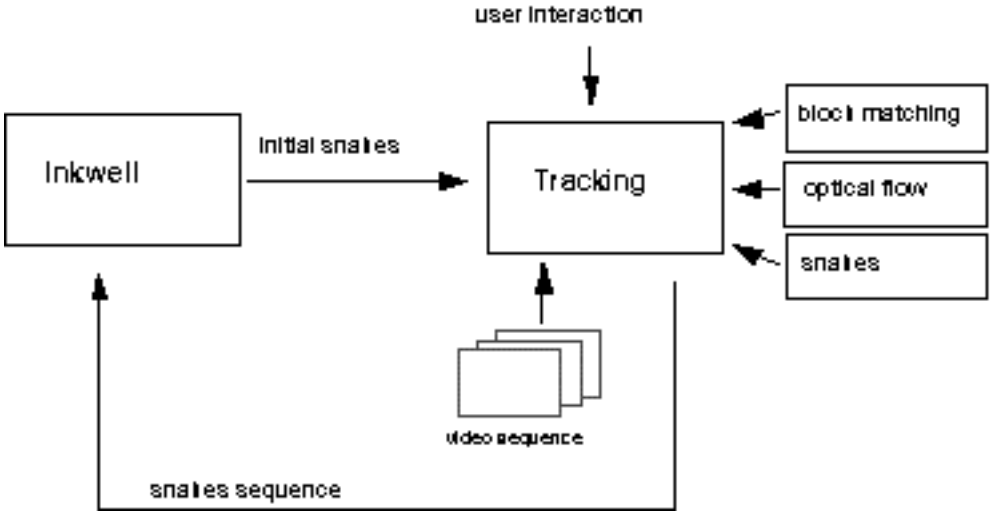


Fig. 2: Block diagram of semiautomatic system

Figure 3 shows the combination of the different techniques. From Fig. 3 it can be seen that the snake finds the correct edge only because it is pushed near it by the optical flow estimate. Otherwise, it would find the adjacent edge which will appear under the snake due to the motion of the underlying object.

Before continuing on to the next frame in a sequence the user may interrupt and fix any errors, relocating the endpoints of a snake or pushing it towards another edge. In our experience, we have been able to track some features as many as 50 or 60 frames without correction (the lip and eyelid edges of the facial sequence, Fig. 4) while others need tending every fourth or fifth frame (the eyebrows and lower jaw line of the same figure). The resulting snake sequence is then read into the animation system and applied to the 2D image of a character (Patterson et al. 1991). Driven by the snake sequence, the animation system performs all necessary morphing on the 2D image to complete the animation.

6 Results

The system has been implemented in C++ on a Silicon Graphics Crimson. It has been applied to a test sequence of facial expressions consisting of approximately 700 frames. Due to the nature of our edge finding algorithm, which quantized to one bit, a median filtering of the tracked motion was done in order to damp the jerkiness of the sequence, yet maintain large discontinuities in the motion when they occurred. A better approach would be appropriate preprocessing of the image sequence before the tracking process begins. After damping, the resulting motion was mapped onto another character using Inkwell, the animation system. The resulting animation reproduced subtle movements of the face (see Fig. 4). The following table show the performance of the system in comparison with a previous manual approach (Inkwell provides an animation rotoscoping facility).

Technique	frames per day	frames tracked
manual approach	20	every 5th frame
semi-automatic system	250	every frame

Table 1: Performance compared to manual approach

The optical flow and nearest edge calculations are independent of the user's interactions and may be performed before the interactive session.

To show the usefulness of this approach for general purpose edge tracking (processing sequences where no special marking or lighting is employed), the system has been used to track features in other video sequences. Two different

outdoor scenes are pictured in Fig. 5: a motorcycle moving in front of a wall with a painted sign and flags in the wind (in Fig. 5, both sequences show every 8th frame from a tracked video sequence). We had no problems tracking the edges that were selected and ambiguities could be resolved by user interaction.

7 Future work

The resulting animations showed good performance for capturing subtle movements of the face in the test sequence, but due to quantization in the edge enhancement phase we derived motion that was much more jerky than need be. Color sensitive snakes could be used for tracking color coded lips that come together and part again, obviating the need for user interaction when the lips part. A motion (or static) model, a la (Blake and Yuille 1992), could also be incorporated into the tracking process which would help reduce errors in the tracking process. We could add knowledge, for example, about what a face can and cannot do when tracking faces. Currently, the motion is tracked in 2D and the animations are morphings of 2D images, but with two or more viewpoints the motion can be tracked in 3D (Patterson, et al., 1991).

8 Conclusion

In this paper we discussed the applicability of snakes to facial feature tracking and outlined the problems that are inherent when snakes are used for tracking multiple frames. We presented a practical solution which combines block matching techniques for tracking the endpoints of the snakes, optical flow to push the snakes towards the target edges, followed by the shape refinement of the snake algorithm itself. A semi-automatic system has been used to process a sequence of facial expressions. Good results have been achieved by using the resulting tracked motion to animate a 2D image of a character, capturing many nuances of the performance. Our solution has also proved useful for general edge tracking. By adding the proposed enhancements to the current implementation an even more automatic system may be created. We showed that the presented technique is useful for performance-driven animation and believe that it bears great potential for other applications that include feature tracking as part of the segmentation or scene analysis phase of processing video sequences.

Acknowledgments

Thanks to Michael Kass, James Normile, Kah-Kay Sung, Lance Williams, and the Dr. Jürgen Schneider Foundation, Frankfurt, Germany.

References

Bergen JR, Hingorani R (1990) Hierarchical Motion-Based Frame Rate Conversion, David Sarnoff Research Center, Princeton NJ 08540.

Blake A, Yuille A (1992) Active Vision, MIT Press.

Burt PJ, Yen C, Xu X (1982) Local Correlation measures for motion analysis, a comparative study. IEEE proceedings, Conference on pattern recognition and image processing, June 14-17, Las Vegas, Nevada.

DeWitt T, Edelstein P (1982) Pantomation: A System for Position Tracking,. IEEE Proceedings of the Symposium on Small Computers in the Arts, October 15-17, pp. 61-69.

Kass M, Witkin A, Terzopoulos D (1988) Snakes: Active Contour Models. International Journal of Computer Vision, 321-331.

Li H, Roivainen P, Forchheimer R (1993) 3-D Motion Estimation in Model-Based Facial Image Coding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15:6, pp 545-555.

Lim JS (1990) Two-Dimensional Signal and Image Processing. Prentice Hall.

Litwinowicz PC (1991) Inkwell: A 2½-D Animation System. Computer Graphics, 25:4, pp 113-122-

Litwinowicz P, Williams L (1994) Animating Images with Drawings. Computer Graphics, SIGGRAPH Conference Proceedings, pp. 409-412.0

Niemann H (1990) Pattern Analysis and Understanding. Second Edition, Springer-Verlag.

Patterson E, Litwinowicz PC, Greene N (1991) Facial Animation by Spatial Mapping,. Computer Animation '91, Spinger-Verlag.

Pratt W (1991) Digitial Image Processing. John Wiley and Sons, Inc.

Terzopoulos D, Waters K (1991) Techniques for Realistic Facial Modeling and Animation. Computer Animation '91, Springer-Verlag.

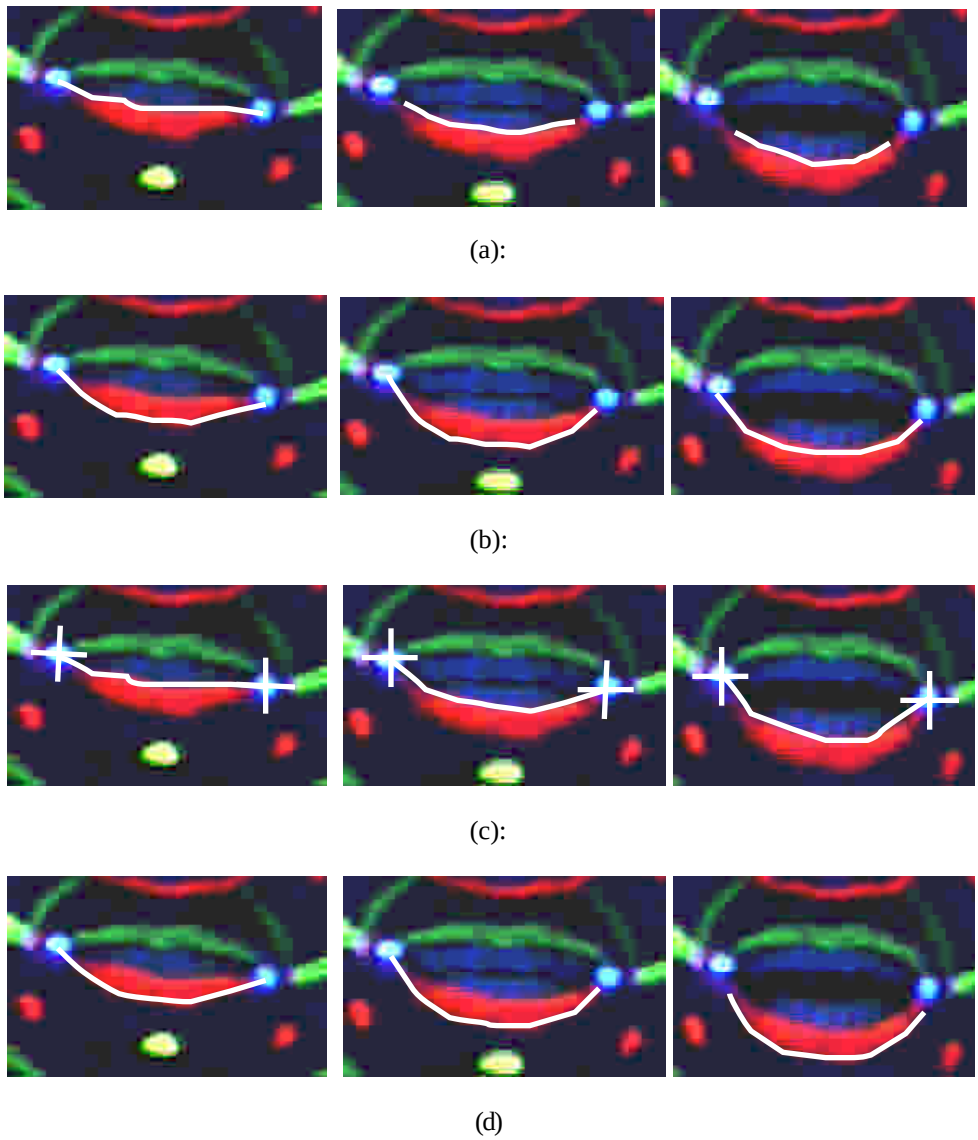
Williams L (1990) Performance-Driven Facial Animation. Computer Graphics, 24,4: 235-242.

Figure Captions

Fig. 4: In the first row an actor with high contrast makeup is shown expressing himself. The second row shows tracked features using the techniques described in the paper. The animated features are used to drive another face using cross-mapping and morphing techniques.

Fig. 5: Using the semi-automatic system described in the paper, features are tracked in non-specially marked video sequences.

© *The Visual Computer, vol. 12, no 2, 1996, pp 75-83.*



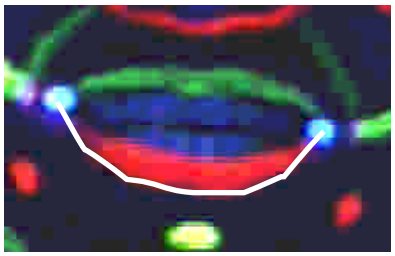
(a):

(b):

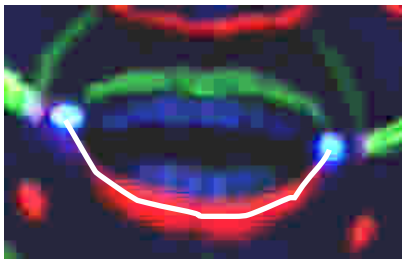
(c):

(d)

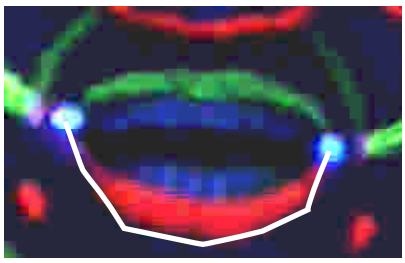
Fig 1: (a) Moving endpoint problem, (b) Snake switching edges due to large object motion, (c) Contribution of block matching, (d): Contribution of optical flow



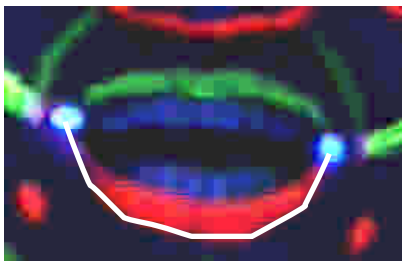
frame 1
the lips of a performer painted with high contrast makeup; the lower lip is tracked by a snake.



frame 2
shown after block matching the ends
Note: Due to the motion of the adjacent edge the snake would find the wrong edge because there are two parallel edges here, and substantial interframe motion, the edge "nearest" the snake in the second frame is not the edge it was supposed to track.



frame 2
shown after block matching the ends and pushing intermediate points via an optical flow algorithm (motion vectors used are slightly exaggerated for demonstration purposes).



frame 2
shown after snake finds its minimum energy
Note: the number of control points with which the edge is modeled and the snake internal stiffness constraint keeps it from following the feature more closely

Fig. 3: Combination of the different techniques