published in IEEE CVPR’99, vol 2, pp 232–238, Fort Collins, June 1999.

DYNAMIC OCCLUDING CONTOURS
A New External-Energy Term for Snakes

Michele M. Covell *and* *Trevor J. Darrell*
covell@interval.com trevor@interval.com

Interval Research Corporation
1801 Page Mill Road, Bldg. C, Palo Alto, CA 94304, USA

Abstract

Dynamic contours, or snakes, provide an effective method for tracking complex moving objects for segmentation and recognition tasks, but have difficulty tracking occluding boundaries on cluttered backgrounds. To compensate for this shortcoming, dynamic contours often rely on detailed object-shape or -motion models to distinguish between the boundary of the tracked object and other boundaries in the background. In this paper, we present a complementary approach to detailed object models: We improve the discriminative power of the local image measurements that drive the tracking process. We describe a new, robust external-energy term for dynamic contours that can track occluding boundaries without detailed object models. We show how our image model improves tracking in cluttered scenes, and describe how a fine-grained image-segmentation mask is created directly from the local image measurements used for tracking.

1 Tracking Boundaries

Tracking visual features in a series of images is an important task both for vision-based control and for rotoscoping applications. Dynamic contours [Kass87], and related active tracking techniques, are well suited for both applications because they combine simple, light-weight object models with rapid updates.

Dynamic contours track boundaries by minimizing the sum of an external force, from a local image measure, and an internal force, from a shape-dynamics model. A dynamic contour tracks the indicated boundary by finding the shape that minimizes the combined external and internal forces. The external force drives the dynamic contour according to the current image appearance. The internal force increases the spatial and temporal continuity of the tracked boundary.

Dynamic contours usually employ a simple image-contrast measure to define the external forces on the model. This approach works well as long as the boundary being tracked is not an occluding boundary, such as that of a silhouette. However, for tasks such as rotoscoping, it is the occluding boundaries that must be tracked.

When the boundary to be tracked is an occluding boundary, the dynamic contour often confuses background texture for the desired boundary. Figure 1 shows a tracking failure when a dynamic contour originally marking the edge of the pen cap instead sticks to the edge of the book.

To help us disambiguate the contrast edges, of course, we could use better models of how books and pens move. Recent work has improved the shape-dynamics models [Cootes93, Blake94, Terzopolous92]. We could also track multiple hypotheses [Isard96] and use future shape distortions to select the correct tracking sequences. These approaches require detailed shape and motion models for each object that we hope to track.

In this paper, we propose an alternative to using simple contrast measures for the external-energy term of dynamic-contour models. Our image model describes the local contrast pattern but is largely insensitive to changes in background contrast. After reviewing previous image models used in tracking in Section 2, we describe this new image model, called radial cumulative similarity (RCS), in Section 3. In Sections 4 and 5, we develop an external-force term based on the RCS transform. In Section 6, we demonstrate the potential of this RCS-based force in tracking and in rotoscoping. Finally, we summarize and conclude in Section 7.

2 Local Image Models

There are two independent choices that we must make when implementing dynamic contours: the nature of the external-force term, which responds to the structure of the image, and the nature of internal-force term, which captures the global shape or dynamics model. Our image model improves the external-force or image-structure term. Therefore, in this section, we review previous work in local image models.

Most dynamic contours use a simple edge-based model as their external-force term [Kass87, Yuille89]. The magnitude of the external force on a node is directly related to the distance between the node and the nearest edge. These edge-based forces are largely invariant to illumination and to background color at occlusions. This invariance is

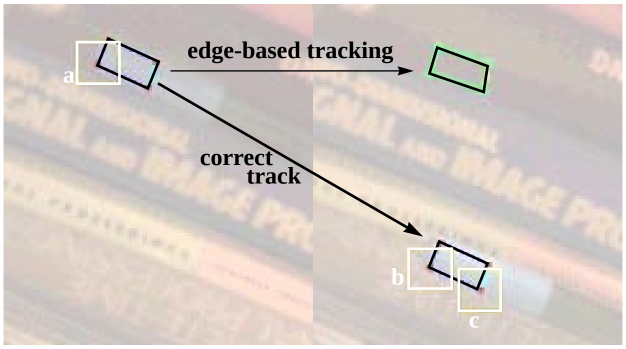


Figure 1: Tracking errors. Edge-based dynamic contours tend to stick to whichever each edges are closest. When the pen moves down rapidly, the edge-based dynamic contour finds the solution that tracks the background edges instead of the pen itself.

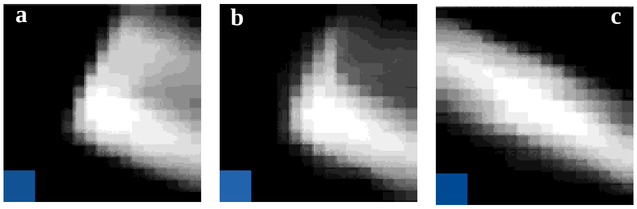


Figure 2: The RCS transforms for points a, b, and c in Figure 1. The color shown in each lower-left corner is the local color for the RCS transform of that point; the remainder of each image shows the similarity pattern for that point.

bought by discarding considerable image information and results in the tracking ambiguities discussed in Section 1.

Image patches are an alternative to an edge-based external force term. To use image patches as the basis for an external-force term, we first associate a desired appearance with each node of the dynamic contour. The external force is then computed according to the vector distance between the desired appearance and the patch underlying each node [Wiskott95] or according to optical flow [Peterfreund97]. This external-force model has the advantage of potentially using all the available color and texture information. It is, however, unreliable near occlusions, due to mismatched background pixels within the image patch. Since one-half of the pixels near occlusions are from the background, the relative motion between the foreground and background has a strong effect on this external-force term.

We can reduce the influence of mismatched background pixels by using robust matching techniques [Black93]. For highly textured foreground objects, dynamic contours using robust matching track the object outline well. However, when the foreground object has little texture, the robust-matching dynamic contour will drift away from the occlusion boundary, toward the object interior. To see why it does, we consider the examples in Figure 1. With robust matching, the match error in Figure 1 from point a to b is essentially the same as the match error from a to c: The same background pixels around in point a are discarded as outliers, and the untextured foreground gives little alignment information.

In summary, simple edge-based measures cannot distinguish between edges, due to lost color and texture information. Matching local image patches suffers from outliers due to changes in background attributes. Matching using robust norms avoids the outlier problem but introduces ambiguities when the foreground object has little texture.

We need an image model that

1. Captures foreground attributes (as does patch matching)
2. Captures the occlusion structure (as does edge matching)
3. Ignores background attributes (as does robust matching)

In Section 3, we describe the matching based on RCS. The RCS transform has all three of these desired image-model properties.

3 Radial Cumulative Similarity

The RCS transform [Darrell98] captures occlusion-boundary information and foreground attributes while ignoring background pixels. It describes each image location using two distinct parts: a local color[1] and a similarity pattern. Figure 2 shows the RCS transform of the three points marked in Figure 1.

The local color provides point information about the image. Its value is not influenced by nearby background regions. Thus, the local color is a reliable—but not a distinctive—description of each location.

The similarity pattern further distinguishes locations from one another by capturing patterns of change in the local color. The similarity pattern highlights occlusion boundaries, while removing the effects of background pixels. Conceptually, it measures similarity between the center color and colors at nearby points, then reduces the effects of similarity values that lie beyond contrast boundaries. This remapping of similarity values reduces the influence of background pixels, since the intervening occlusion boundary will typically be marked by a contrast boundary. Furthermore, the remapping creates a rapid transition, from high to low similarity, which actually highlights occlusion boundaries (as well as other contrast boundaries).

This approach is formalized in the following definitions and equations. First, we define a mixed-indexing notation. Given a 2D image $\mathbf{I}$, whose values are normalized to between 0 and 1, $\mathbf{I}_{x_0, y_0}(r, \theta)$ is the image value that is r units away from (x_0, y_0), in the direction θ.

The RCS transform associates with each image location, (x, y), a local color, $\mathbf{C_I}(x, y)$, and a similarity pattern, $\mathbf{N_I}(x, y, r, \theta)$. The local color, $\mathbf{C_I}(x, y)$, is simply the image color smoothed over a compact region. We define the similarity pattern, $\mathbf{N_I}(x, y, r, \theta)$, by using point dissimilarities.

The point dissimilarity, $\mathbf{D_I}(x, y, r, \theta)$, measures the distance between the local color, $\mathbf{C_I}(x, y)$, and the image values, $\mathbf{I}_{x, y}(r, \theta)$:

$$\mathbf{D_I}(x, y, r, \theta) = \alpha^2 \left\| \mathbf{C_I}(x, y) - \mathbf{I}_{x, y}(r, \theta) \right\|^2 .$$

Note that the point dissimilarity, $\mathbf{D_I}(x, y, r, \theta)$, depends on only the magnitude, and not on the direction, of the color change.

We compute the similarity pattern, $\mathbf{N_I}(x, y, r, \theta)$, from the point dissimilarities by propagating $\mathbf{D_I}(x, y, r, \theta)$ radially. Specifically,

$$\mathbf{N_I}(x, y, r, \theta) = \exp\left\{ -\int_{\rho < r} d\rho \; \mathbf{D_I}(x, y, \rho, \theta) \right\}$$

Once the similarity pattern encounters a large dissimilarity, it reduces the influence of the similarity values farther along that ray, by whitening them toward 0. The point dissimilarity on the background side of the occluding boundary is expected to be large due to its arbitrary changes from the foreground color. Cumulative integration and negative

[1] We use color in our descriptions of RCS for the remainder of this paper. Other image properties, such as texture, can also be used as the basis of RCS transforms.

exponentiation lessens the influence of these background fluctuations: The larger the intervening dissimilarities, the less background patterns affect the RCS similarity.

With isolated points, matching using RCS transforms outperforms matching using conventional L2 or Lorenzian norms on several difficult cases [Darrell98]. In the remainder of this paper, we describe modifications to the RCS transform, for use in occlusion tracking and in rotoscoping.

4 Dynamic Occluding-Contour Models

This section describes three ways to use RCS transforms: (1) as the external-force term in a sparsely sampled version of dynamic contours, (2) as the external-force term in a densely sampled version of dynamic contours, and (3) as an object-profile description in automatic rotoscoping.

4.1 Sparse-RCS Dynamic Contours

In its simplest form, tracking using RCS dynamic contours is similar to matching isolated RCS samples [Darrell98]. First, we associate each node of the dynamic contour with a desired RCS value, taken directly from the image pattern at the inital nodal points of the user-drawn contour. The external force in subsequent frames is then computed according to the vector distance between the desired RCS and the RCS of the proposed node location.

This approach to RCS dynamic contours is fast, since only sparse RCS samples (those at the nodal points) are needed and since the external force at each node is independent of the locations of the other nodes. This approach to image matching outperforms isolated-point RCS matching, since the internal-energy term helps to disambiguate alternative matches.

There are two shortcomings to the sparse-RCS dynamic contour. The first is the lack of rotational invariance: The RCS coordinates are tied to the image-plane coordinates, instead of rotating with the rotation of the dynamic contour. We can correct this shortcoming by resampling the similarity patterns, using the rotation implied by the new contour orientation. A second, more fundamental, shortcoming is the sparseness of RCS samples that the dynamic contour uses to determine the external forces. If the nodal points happen to fall in an extremely low- or extremely high-contrast area, the very sparseness that makes this dynamic contour fast makes it vulnerable to false minima.

Even with these shortcomings, sparse-RCS dynamic contours are well suited for matching over large areas, such as the search windows within isolated images or coarsely sampled sequences. The sparse-RCS dynamic contour often gives a good coarse level match, which we can refine using a dense-RCS dynamic contour.

4.2 Dense-RCS Dynamic Contour

We can avoid the shortcomings of the sparse-RCS dynamic contour by using a ray from each RCS transform along the full length of the dynamic contour. This sampling of the RCS provides rotational invariance, since we use rays that are perpendicular to the local contour orientation. It also avoids the ambiguities introduced by matching only nodal points, since the external-force term is determined by the RCS appearance along the full length of the contour.

The RCS profile between (x_0, y_0) and (x_1, y_1) has two parts: a local-color line, $\mathbf{C_I}(s|x_0, y_0, x_1, y_1)$, and a similarity profile, $\mathbf{N_I}(s, r|x_0, y_0, x_1, y_1)$. The local-color line is simply the previously defined local color, sampled on the line segment from (x_0, y_0) to (x_1, y_1). The argument s indexes different locations along the line segment. The local-color line is given by

$$\mathbf{C_I}(s|x_0, y_0, x_1, y_1) = \mathbf{C_I}(x_s, y_s)$$

where $x_s = x_0(1-s) + x_1 s$ and $y_s = y_0(1-s) + y_1 s$.

The similarity profile contains samples from the RCS similarity patterns. The RCS similarity patterns are sampled along the line segment, then these selected similarity patterns are themselves sampled along the angle perpendicular to that line segment:

$$\mathbf{N_I}(s, r|x_0, y_0, x_1, y_1) = \mathbf{N_I}(x_s, y_s, r, \theta_\perp)$$

where x_s and y_s are as before, and $\theta_\perp$ is the left-perpendicular angle to the vector $\{(x_1 - x_0), (y_1 - y_0)\}$.

We associate with each node of the dynamic contour a desired RCS profile, taken along the line segments between the current node and two neighbors. The external force in subsequent frames is the vector distance between the desired RCS profile and the RCS profile, using the proposed node locations.

The dense-RCS dynamic contour provides rotational invariance and avoids local minima that can distract the sparse-RCS dynamic contour. It handles rotations, since the axes of the similarity profiles are defined relative to the nodal locations. It avoids local minima that can trap the sparse-RCS dynamic contour, since it relies on the match error along the full length of the contour, instead of just at the nodal points. Also, the external-energy term of the dense-RCS dynamic contour accommodates stretching (and shrinking) of the contour, since the sample density of the RCS profile is normalized by the segment length. This property is useful for matching certain types of nonrigid deformations, such as the stretching that occurs along the edge of the lips as the mouth opens.

The dense-RCS dynamic contour is well suited for tracking in sequences. In this case, we use the contour configuration from the previous frame to estimate the position in the current frame. To avoid accumulation of positioning errors, when the matching error exceeds a given threshold, we archive the current RCS profile and then update the profile being used. The updated profile is a previously archived profile, if there is an archived profile with a low enough matching error; if not, a new profile is created from the patterns in the current image [Huttenlocher93].

The dense-RCS dynamic contour is well suited for refining a coarse match given by a sparse-RCS dynamic contour. By initially using the sparse-RCS match, we reduce the search area needed for the dense-RCS dynamic contour. The dense-RCS dynamic contour then improves the detailed alignment along the full length of the contour.

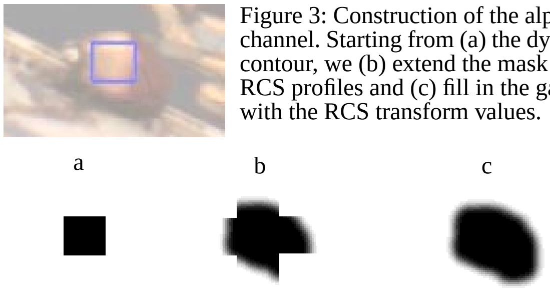


Figure 3: Construction of the alpha channel. Starting from (a) the dynamic contour, we (b) extend the mask using RCS profiles and (c) fill in the gaps with the RCS transform values.

4.3 RCS Rotoscoping

RCS dynamic contours, along with their associated RCS profiles, can be used for semi-automatic rotoscoping of foreground objects. The artist draws a dynamic contour in the first frame of the sequence. The drawn contour should lie fairly close to the outer boundary of the object, but with the full length of the contour completely inside the object's occluding boundary. We then close the dynamic contour by logically connecting the first and final nodes to make them neighboring nodes.

We propagate this closed dynamic contour through the entire sequence using dense-RCS dynamic contours or, if the temporal sampling is coarse, using sparse-RCS dynamic contours (to update) followed by dense-RCS dynamic contours with a smaller search area (to refine). Then, we rotoscope the selected object out of the sequence automatically, using a continuous-valued alpha-channel sequence created from the contour position and from the RCS profiles along the contour.

The alpha channel for each frame is initially set to 1 inside the dynamic contour and to 0 outside (Figure 3-a). The alpha-channel values are then modified according to the RCS profile values. Taking each line segment of the dynamic contour in turn, we set the alpha-channel value at each location to the maximum similarity-profile value (Figure 3-b). Finally, we increase the alpha-channel values near the contour nodes according to the RCS similarity patterns for the nodal point, so as to fill in the sharp angles that otherwise occur at convex contour nodes (Figure 3-c).

This process extends the edges of the selected region to the nearest edge in the image. This extension softens the hard edges of the selected region providing a smooth roll off. It also allows the edges of the selected region to curve according to the details of the object's shape, instead of being formed from straight line segments.

The resulting rotoscoping masks may be too large if the background region near the object's outer boundary is similar in color to the object itself, such as the colors of the lips and gums. The dense-RCS dynamic contour does not mistake these regions for object points, since it integrates the match error along the full length of each line segment. To create the alpha channel using a similar spatially integrating philosophy, we could modify the alpha channel using the 1D average of the RCS profile, instead of using the full 2D appearance. This 1D average would still allow the boundaries of the alpha-channel mask to roll-off gradually. Unfortunately, it would also create mask edges that are a series of straight line segments, rather than the curved object boundaries that we want.

Instead of using spatial integration, we use temporal integration to keep the rotoscoping mask from overexpanding. In particular, we construct the alpha-channel values using the minimum values of the RCS profile images across the consecutive frames. We then use this (temporal-)minimum similarity profile to form the alpha-channel mask.

RCS rotoscoping is appropriate for most rotoscoping applications. In addition, we can use these alpha-channel masks to separate disconnected layers of an image, prior to morphing [Litwinowitz94].

5 Implementation Details

We implemented our dynamic contours using a dynamic-programming method [Amini90]. We modified the basic implementation of this method [Bregler95] to find optimal solutions for closed contours: We added a layer of deferred decisions to account for the different last-node positions. This modification increases the memory requirements but does not significantly increase the computational requirements.

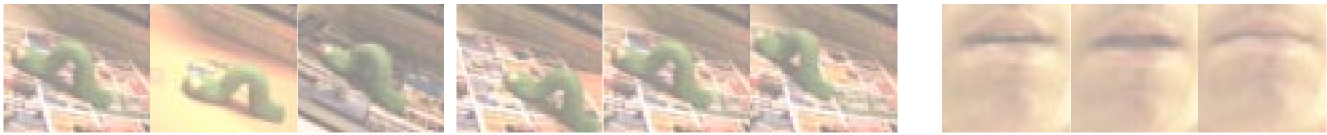
As described in Section 4, the external-energy term of our RCS dynamic contours uses the match error between the current and the desired RCS transforms:

$$E_E(S|\mathbf{I},\mathbf{I}_d) = \sum_n \Delta\mathbf{C}(n|S,\mathbf{I},\mathbf{I}_d) + \lambda\Delta\mathbf{N}(n|S,\mathbf{I},\mathbf{I}_d)$$

where $\Delta\mathbf{C}(n|S,\mathbf{I},\mathbf{I}_d)$ and $\Delta\mathbf{N}(n|S,\mathbf{I},\mathbf{I}_d)$ are the color and the similarity errors, respectively, associated with the *n*th node under the current contour configuration, S. For sparse-RCS dynamic contours, the color error is the sum squared difference between the local color in image **I** under the *n*th node and the desired local color for the *n*th node. The similarity error is the mean, over all radii below R_N, of the squared difference between the similarity pattern in image **I** under the *n*th node and the desired similarity pattern for the *n*th node. For dense-RCS dynamic contours, the color error is the sum, over the segment, of the sum squared difference between the color profile in image **I** and the desired color profile, both on the line segment from the *n*th-node to the (*n*+1)th node. The similarity error is the mean, over all radii below R_N, of the squared difference between the similarity profile in image **I** and the desired similarity pattern, both on the line segment from the *n*th node to the (*n*+1)th node. For all our RCS dynamic contours, R_c=3, α=10, and R_N=12.

We compare our tracking performance with that obtained with edge-based and patch-based dynamic contours. For both the latter, we define the external energy term using the evidence integrated over the full contour.

In our edge-based dynamic contour, the external energy is the sum, over the segment, of the squared Laplacian of the image **I**, on the line segment from the *n*th node to the (*n*+1)th node. In our patch-based dynamic contour, the external energy is the sum, over the image patch, of the squared differences between the image **I** and the desired appearance $\mathbf{I}_d$ on the rectangular image patch taken along



(a) Related stills (b) A sequence with coarse temporal sampling (c) A sequence with fine temporal sampling (nonconsecutive frames are shown).

Figure 4: Examples of the types of frames used in testing

the line segment from the *n*th node to the (*n*+1)th node and extending to a distance R_N to either side of the line segment.

The internal-energy term, $E(S|S_0)$, for all of our dynamic contours is a simple second-order model of shape dynamics:

$$E_I(S|S_0) = \sum_n (d(n|S) - d(n|S_0))^2 + \beta(c(n|S) - c(n|S_0))^2$$

where S_0 and S are the previous and current contour configurations, $d(n|S)$ is the distance between the *n*th and the (*n*+1)th nodes, and $c(n|S)$ is related to the curvature at the *n*th node. Specifically, $c(n|S)$ is the minimum distance between the *n*th node and the line segment connecting the (*n*–1)th and the (*n*+1)th nodes.

We combined linearly the internal- and external-energy terms of the dynamic contours to get the total energy of the dynamic contour. For simplicity, we used the same combination weights at all nodes.

6 Results

We tested our tracking and rotoscoping on three types of input samples (Figure 4): related stills, a coarsely sampled sequence; and a finely sampled sequence.

The related stills and the coarsely sampled sequence show large displacements in a cluttered environment. For these images, we used the sparse-RCS followed by the dense-RCS, as outlined in Section 4.2. Our results from the stills are shown in Figure 5. Samples from the coarsely sampled sequences are shown in Figure 6.

The finely sampled sequence shows a low-texture, deformable occlusion (the inside boundary of the lips), with a similarly colored background (the teeth, gums, and the opposite lip) along a related but distinct path. We tracked the finely sampled sequence using dense-RCS dynamic contours with template updating. These results are shown in Figure 7.

The corresponding rotoscoping results are shown in Figure 8.

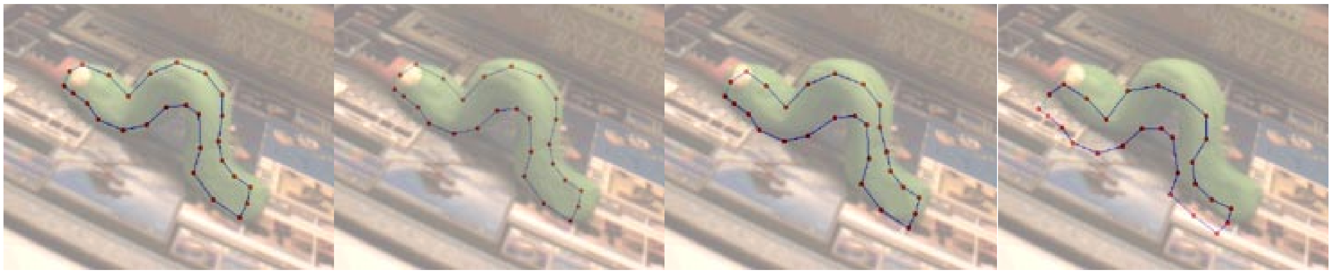
6.1 Tracking Results

The RCS dynamic contours performed best on the cluttered-background tests (Figures 5 and 6). In these cases, there is a consistently high contrast across the occluding boundary. The dynamic contour tended to drift more on the mouth sequence (Figure 7), particularly when the lower gums were visible. Even so, the RCS dynamic contours remained attached to the lips throughout the sequence. It archived four updates over the course of 115 frames, and reused two of the archived patterns (for a total of six updates in 115 frames).

The patch-based dynamic contour did well at matching still images (Figure 5): The dynamic contour was far enough inside the object boundary that the (background) outliers within the match integral did not throw it too far off. However, for coarsely and finely sampled sequences, the patch-based dynamic contour did not track well. Its tracking errors in the coarsely sampled sequence accumulated over the sequence, distorting the expected shape of the contour. On the finely sampled sequence, the patch-based dynamic contour tracked the innermost boundary of either teeth or lips, instead of remaining attached to the lips when the teeth appeared.

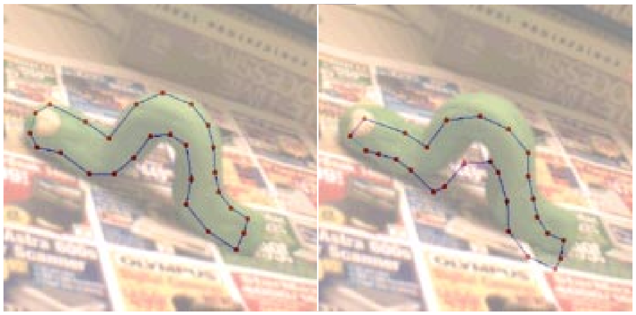
As expected, the edge-based dynamic contour gave poor matches when confronted with a cluttered background (Figure 5). This poor performance continued, even when the edge-based dynamic contour was initialized near the correct solution and, more surprising, even in the weakly textured mouth example. In light of this consistently poor performance, we omitted the edge-based dynamic contour results from Figures 6 and 7.

The improvements provided by the RCS dynamic contours do not accrue without a cost: The RCS external energy model is more expensive computationally than are the classic edge- and patch-based dynamic-contour models.



dense RCS sparse RCS patch-based edge-based

Figure 5: Dynamic-contour tracking between still frames. The reference for RCS and patch-based tracking was taken from the frame shown at the right.



dense RCS patch-based

Figure 6: Dynamic-contour tracking on a coarsely sampled sequence.

In sparse-RCS dynamic contours, the computational complexity of the external energy term is O(NMR^2), where N is the number of nodes in the contour, M is the search area in the target image, and R^2 is the size of the RCS similarity pattern. With dense-RCS dynamic contours, the complexity can be as high as O(LM^2R), where L is the total length of the contour. This increased complexity results from the dense RCS sampling (changing the dependence from N to L) and from the dependence of the external force at each node on the neighboring nodes (changing the dependence from M to M^2). Caching and interpolating samples of the similarity profiles reduce the computational complexity to O(LMR^2). Finally, if speed is essential, we can reduce the computational complexity to O(LMR) by using a single, nominally perpendicular angle for each inter-nodal segment, as well as using caching.

6.2 Rotoscoping Results

We also used the dense-RCS dynamic contours for semi-automatic rotoscoping. The results are shown in Figure 8.We include both halves of the rotoscoping result: The foreground piece shows where the rotoscoping mask was too large, and the background piece shows where it was to small.

Again, as expected, the rotoscoping process worked especially well on the high-contrast backgrounds (the stills and the coarsely sampled sequence). Its tendency to clip just inside the object's outer boundary, instead of exactly on it, is useful if the foreground object is the thing that is kept, but it leaves a small rim around the hole in the background sequence. In contrast, on the mouth sequence, the rotoscoping mask tended to associate too much of the interior of the mouth (the background) with the surrounding lips (the foreground). Most of this over-inclusion was removed by the temporal integration described in Section 6.2.

7 Conclusions

We have presented a new external-energy model for dynamic contours, based on the RCS profile, which is both stable and distinctive at object's outer boundaries. Using this model, RCS dynamic contours tracked occluding boundaries in cluttered scenes, with the simplest of internal-energy terms. The RCS profile is also useful in semi-automatic rotoscoping: The artist marks the object outline just once, in only the first frame of the movie sequence. From that starting position, the RCS dynamic contours track the outline in subsequent frames. Finally, the rotoscoping mask itself is tailored to the detailed outline of the object by the RCS profile. The strength of our external-energy model is illustrated by the results shown in Figures 5 through 7. The quality of our rotoscoping method is illustrated in Figure 8.

References

A. Amini, T. Weymouth, and R. Jain. "Using Dynamic Programming for Solving Variational Problems in Vision," *IEEE Trans. PAMI* 12(9): 855–867, 1990.

M. Black and P. Anandan. "A Framework for Robust Estimation of Optical Flow," *Proc. ICCV,* pp. 263–274. Berlin, Germany, 1993.

A. Blake and M. Isard. "Three-Dimensional Position and Shape Input Using Video Tracking of Hands and Lips," *Proc. SIGGRAPH'94,* pp. 185–192, Orlando, FL, 1994.

C. Bregler and M. Slaney. "Snakes," IRC–TR 1995-017, Interval Research Corporation, Palo Alto, CA, 1995. (see http://web.interval.com/papers/1995-017/)

T. Cootes, C. Taylor, A. Lanitis, D. Cooper, and D. Graham. "Building and Using Flexible Models Incorporating Grey-Level Information," *Proc. ICCV,* pp. 242–246, Berlin, Germany, 1993.

T. Darrell, "A Radial Cumulative Similarity Transform for Robust Image Correspondence," *Proc. CVPR,* Santa Barbara, CA, 1998.

D. Huttenlocher, J. Noh, and W. Rucklidge. "Tracking Non-rigid Objects in Complex Scenes," *Proc. ICCV,* pp.

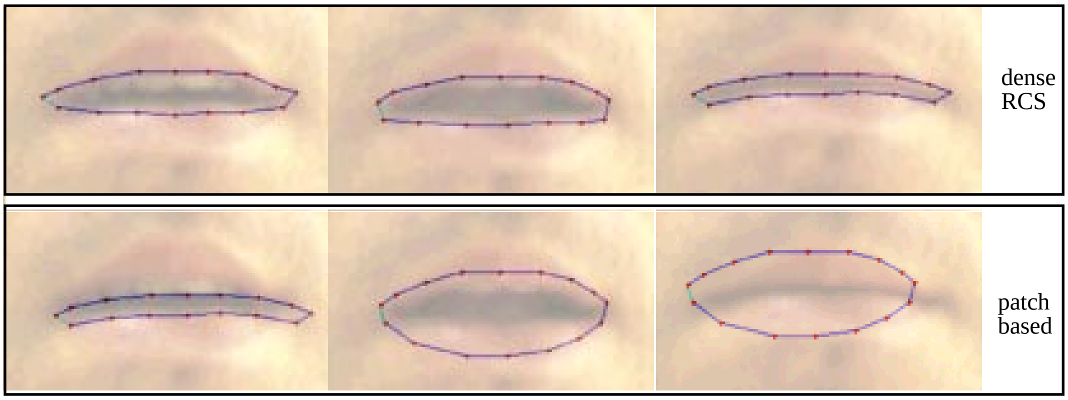


Figure 7: Dynamic-contour tracking on finely sampled sequences (frames 45,107, and 111 are shown here)

93–101, Berlin, Germany, 1993.

M. Isard and A. Blake. “Visual Tracking by Stochastic Propagation of Conditional Density,” *Proc. ECCV*, pp. 343–356, Cambridge, England, 1996.

M. Kass, A. Witkin, and D. Terzopolous. “Snakes: Active Contour Models,” *Proc. ICCV*, pp. 259–268, London, England, 1987.

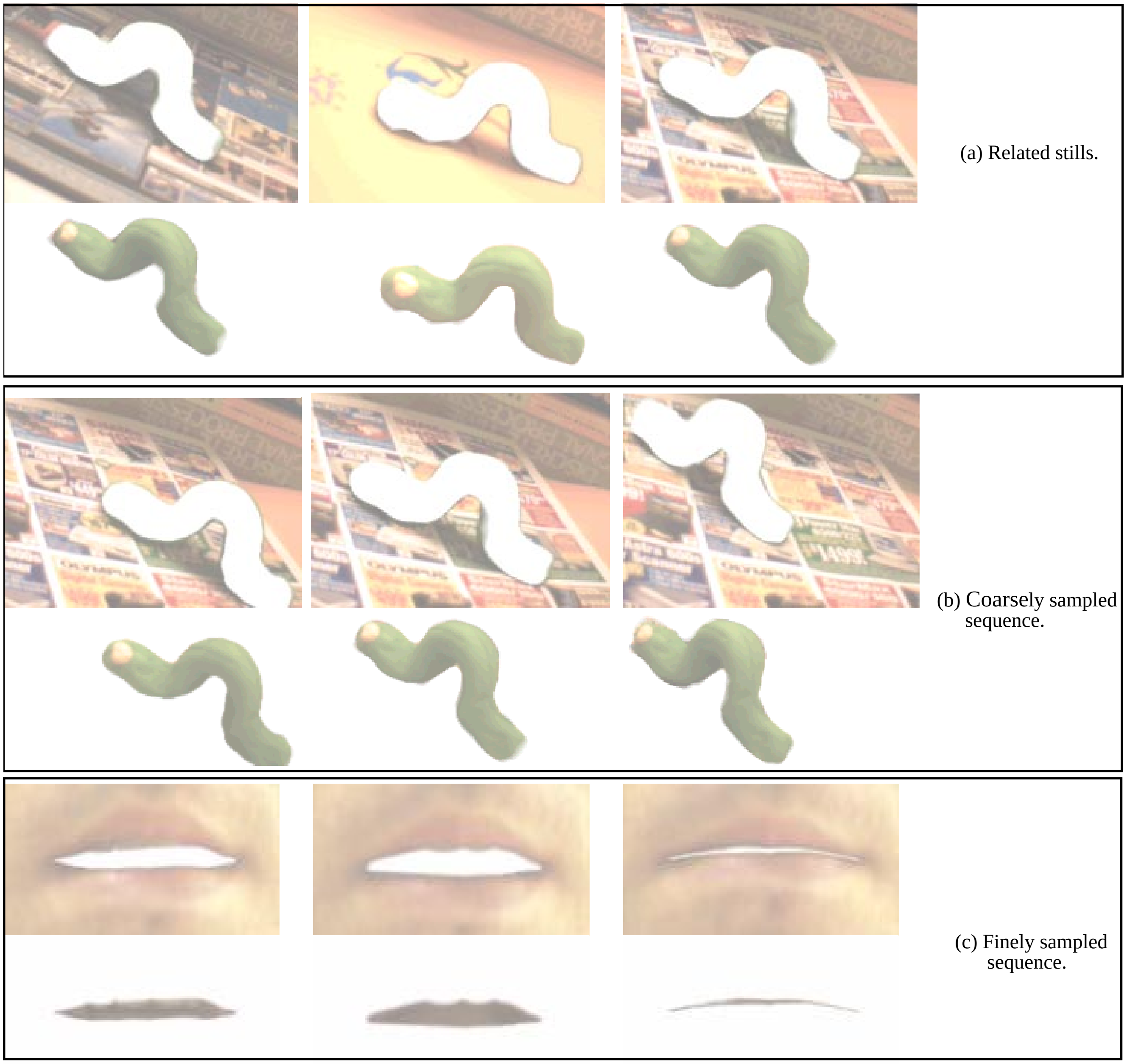
P. Litwinowicz and L. Williams. “Animating Images with Drawings,” *Proc. SIGGRAPH*, pp. 409-412, Orlando, FL, 1994.

N. Peterfreund. “The Velocity Snake,” *Proc. IEEE Nonrigid and Articulated Motion Workshop*, pp. 70–79. San Juan, Puerto Rico, 1997.

D. Terzopolous and R. Szeliski, “Tracking with Kalman Snakes,” in *Active Vision* (eds. A. Blake, A. Yuille), pp. 3–20, MIT Press, Cambridge, MA, 1992.

L. Wiskott, J. Fellous, N Kruger, and C. von der Malsburg, “Face Recognition and Gender Determination,” *International Workshop on Automatic Face and Gesture Recognition*, pp. 92–97. Zurich, Switzerland, 1995.

A.L. Yuille, D.S. Cohen, and P.W. Hallinan, “Feature Extraction from Faces Using Deformable Templates,” *Proc. CVPR*, pp. 104-109. San Diego, CA, 1989.



(a) Related stills.

(b) Coarsely sampled sequence.

(c) Finely sampled sequence.

Figure 8: Results from automatic rotoscoping. We generated these examples using RCS dynamic contours (Sections 4.1 and 4.2) to track and using approach described in Section 4.3 to create the alpha-channel.