Tracking of Tubular Molecules for Scientific Applications[1]

B. Parvin, C. Peng, W. Johnston and M. Maestre
Information and Computing Sciences Division
Lawrence Berkeley Laboratory
Berkeley, CA 94720

Abstract

In this paper, we present a system for detection and tracking of tubular molecules in images. The automatic detection and characterization of the shape, location, and motion of these molecules can enable new laboratory protocols in several scientific disciplines. The uniqueness of the proposed system is twofold: at the macro level, the novelty of the system lies in the integration of object localization and tracking using geometric properties; at the micro level, in the use of high and low level constraints to model the detection and tracking subsystem. The underlying philosophy for object detection is to extract perceptually significant features from the pixel level image, and then use these high level cues to refine the precise boundaries. In the case of tubular molecules, the perceptually significant features are anti-parallel line segments or, equivalently, their axis of symmetries. The axis of symmetry infers a coarse description of the object in terms of a bounding polygon. The polygon then provides the necessary boundary condition for the refinement process, which is based on dynamic programming. For tracking the object in a time sequence of images, the refined contour is then projected onto each consecutive frame.

1 Introduction

Object detection and tracking has been a major area of research in the computer vision community. Typical applications include target tracking, autonomous navigation, process control, and scientific data analysis. In this paper, we focus on a specific tracking problem for scientific applications. The main thrust in such scientific applications is either to gain new insights about the behavior of an object or to establish-and-verify certain hypotheses [16, 5]. We constrain our domain to objects that are elongated, maintain parallel symmetries, and are terminated at both ends. In this paper, we illustrate the performance of our approach through an implementation that detects and tracks DNA molecules (a class of tubular objects) obtained through epi-fluorescence microscopy [4]. An epi-fluorescent microscope is a typical scientific imaging device that operates at the extreme limits of the optical system and at the photo conversion limits of the receptor system [16, 5, 4].

The main novelty of our system lies in its use of high level geometric features for object detection and the subsequent use of this information in the localization of individual molecules, followed by the re-use of this information for tracking. High level constructs are extracted from *incomplete* low level information, and then used to refine and *complete* the low level results. In the application domain, the high level constructs, or the *most perceptually significant* features, are ribbon-like structures defined by a collection of substructures called U-shapes and anti-parallel segments. These isolated segments are grouped with respect to the object model where a coarse description of the molecule is obtained in terms of a bounding polygon. This global representation is then refined using local pixel

[1] Research was supported by the U.S. Dept. of Energy, Office of Scientific Computing, under Contract DE-AC03-76SF00098, and in part by NIH, under grant number AIO8427-25.

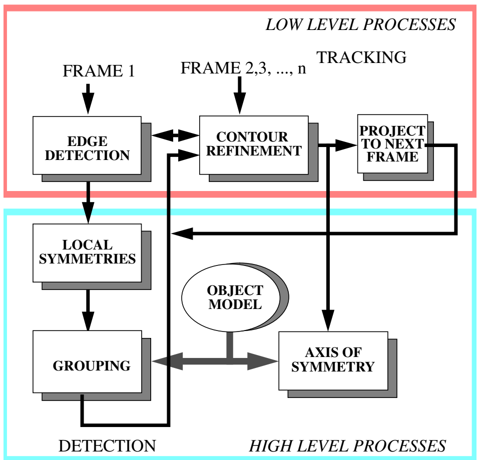


Figure 1: System architecture: the detection is performed only in the first frame to localize the approximate boundaries of individual molecules.

activities. The refinement is achieved through dynamic programming that ensures global optimality. The refined contour is then projected and updated in every consecutive frame. In this context, the detection is only performed in the first frame as an initialization scheme. Furthermore, to facilitate subsequent quantitative analysis, the refined contour is represented by its axis of symmetry. This representation is then used to study the motion and mechanical properties of the molecules. The general architecture of this system is shown in figure 9, and consists of detection, tracking, and shape representation subsystems. In the next section, we briefly review the relevant work in the area, and point out to the similarities as well as unique aspects of our system with respect to previous work. Then, in sections 3 and 4, we outline the details of the detection and tracking subsystems and provide examples to demonstrate their performance on real data. Finally, we conclude our paper in section 5 by discussing the strengths and weaknesses of our approach.

2 Past work

Most of the previous papers in the vision literature aim at either detection *or* tracking of objects of interest in the scene. However, very little attention has been given to combining these techniques for real world applications. From this perspective, several papers have dealt with detection of ribbons [8], generalized cones [3], and more recently, generalized tube model [7]. A brief review follows.

Huertas, et al., [8] developed a system to detect runways in aerial images. They used a hypothesis-and-verification strategy to achieve their goal. The initial hypotheses are obtained by detecting peaks in the histogram of edge orientation images. The underlying idea is that such man-made objects will have a *global* orientation. *Local* hypothesis formation is initiated by detecting anti-parallel segments from low level edge information. These isolated anti-parallel segments, referred to as *apars*, are then joined based on their continuity, collinearity, and consistency with global orientation and certain knowledge about airport engineering. Brooks [3] developed a system called ACRONYM, which has been influential in several vision systems. A typical application of this system is to identify airplanes in aerial images of airfields. In this system, objects are modeled as subpart hierarchies in terms of generalized cones (which are two dimensional shapes, and consist of a cross section swept across a spine). More recently, Huang and Stockman [7] introduced the generalized tube model, where the tube model is locally similar to GC. Their tube model combines contour- and shading-based techniques in a unique fashion. The contour information is used to hypothesize local tubes that are verified by matching their shading properties with dynamically generated optimal filters. The recognition step then sweeps across locally verified tubes using a best fit constraint.

In the context of tracking, most of the previous work is based on a variation of the *Snake* model [9]. This model assumes that an initial contour, defined interactively, is placed near the desired boundaries by means of a closed polygon or spline. This initial contour is then refined using a cost function that integrates internal and external energies, i.e.,

$$E_{snake} = \alpha E_{external} + \beta E_{internal} \tag{1}$$

where the external energy is measured by image forces such as gradients or zero crossings, and the internal energy is measured by the amount that the contour is being stretched or bent. Several refinement strategies based on relaxation [17], variational calculus [9, 10], and dynamic programming [2] have been proposed. The relaxation technique optimizes a local cost function, where the other two aim at global optimization of the cost function. There are some numerical anomalies in the variational approach that can be side-stepped by the other two techniques. Recent approaches to tracking of non-rigid objects [13, 14, 11, 1] assume that the object motion is not totally unstructured and can be modeled based on the physical properties of the object. However, in our particular application domain, i.e., tracking of molecules, such parameters as mass, damping and stiffness are not always known in advance. Furthermore, these parameters are also affected by the medium in which the molecules are being studied, i.e., viscosity of the fluid and the strength of the external stimulus.

Our system operates on the 2D flat world, and in some ways has been influenced by the work of Brooks [3] and Huertas [8]. Its similarity to ACRONYM is in the use of an object model to drive the search process. And its commonality with Huertas's work is in the use of high level features for grouping. Still, our formulation for the particular application is unique in terms of the consistent labeling of the high level features. Yet, because of the fluorescent nature of our imaging modality, the shading information cannot be used effectively as it has been in [7]. The detection phase of our system provides a coarse approximation, in terms of a bounding polygon, for each molecule. The fact that this description is very close to the actual boundary of the molecule, and that the motion of each molecule between consecutive frames is only several pixels, has enabled us to design a simple tracking mechanism that significantly limits the scope of search during refinement. Furthermore, our cost function is simpler than traditional Snake model; it does not rely on the rectangular tessellation of the pixel layout to compute the corresponding energy function, and it eliminates the scaling parameters for the internal and external forces.

3 Detection subsystem

In this section we summarize different components of the detection process as shown in figure 9, and we present intermediate results for each important step of the process. The edge detection is based on Canny's approach [6], which is inherently a gradient operator. The resulting edges are linked, curve segments are extracted, and polygon representation of these curve segments are obtained. The edge gradient and directions are computed at a coarse and fine scale. The coarse scale information is used in the detection phase for better noise immunity, and the fine scale information is used in the tracking phase for better localization. In general, due to noise and variation in contrast, the edge detection technique produces broken and undesirable curve segments. The objective is to group these curve segments such that the individual objects can be extracted from background. This is accomplished by representing curve segments by richer representations which are referred to as "ribbons" in the computer vision literature. A ribbon is a polygon formed by a set of anti-parallel segments or U-shapes. These local ribbons correspond to local symmetries that provide the foundation for grouping them into a larger structure based on a suitable object model. The ribbon representation is difficult to compute, but it simplifies the process of high level reasoning.

3.1 Local symmetries

Local symmetries are identified either from U-shapes or antiparallel segments. U-shapes correspond to the ends of a molecule or points of maximum curvature, and antiparallel segments correspond to the main body of the molecule.

An anti-parallel line is a pair of parallel line segments with opposite directions that can be projected onto one another. This construct is sometime referred to as an "apars" in the literature [8]. There are two types of anti-parallel lines in our system. These are the anti-parallel lines formed by line segments that belong to the *same* curve segment, and the anti-parallel lines that are formed by the line segments from *different* curve segments. The reason for this partitioning is that anti-parallels formed by the same curve segment also infer a U-shape. If the anti-parallel segments occur between different pieces of the same curve segments then the search process (for anti-parallel) is initiated from the position of the U-shape. On the other hand, if an anti-parallel segment occurs between different curve segments, then they can only be discovered by scanning normal to a line segment and forming a correspondence histogram as suggested in [8].

3.2 Grouping

We express the grouping problem as a function of three entities:

$$DNA = Group(features, object\ model, geometric\ constraints) \tag{2}$$

In this formulation, *features* correspond to nodes with labels, and some attributes where a label can be either a U-shape or anti-parallel, and the attributes that encode its invariant geometric measures. The *object model* represents the relationship between the nodes of an idealized DNA model as described by U-shapes and anti-parallels. The *geometric constraints* encode the rules by which the nodes can be linked together. The goal of the grouping is to connect high level features, represented as nodes of a disconnected graph, in such a way that labeling of these nodes remains consistent with respect to the object model subject to the geometric constraints. In this approach, the grouping function is self-correcting since a single node might be mislabeled due to various perturbations in the lower level processes, and errors can be corrected as the grouping process continues. We now provide a summary description of the object model, geometric constraints, and

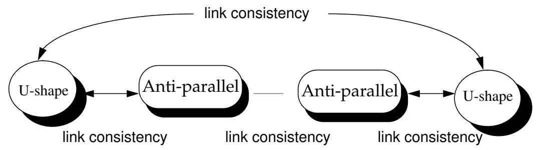


Figure 2: The model for a tubular molecule

the grouping strategy to link a disconnected graph in a coherent fashion. More details can be found in [12].

3.2.1 A model for tubular molecule

The object model is based on our a priori knowledge about the occurrence of objects in an image that may include shape, texture and context. A tubular molecule is distinguished by its symmetric-elongated shape with the following features: i) every DNA has two U-shapes, one at each end; ii) there are anti-parallel lines between these two U-shapes; iii) the variation in the width of a DNA is small; and iv) if we represent the U-shapes and anti-parallels as nodes, and the relationships between them as a link, then the links must be consistent. We refer to this as "Link Consistency". We represent the relationship between the nodes of a DNA object as a graph as shown in figure 10.

3.2.2 Geometric constraints

The goal of the grouping process is to connect high level features in such a way that labeling of these features remains consistent with the geometric constraints of the object model. These geometric constraints can be described as:

1. *Multiple-overlapping ribbons cannot be formed by the same line segments:*
   This constraint is implied by the DNA model, and is used to filter out irrelevant high level features and reduce the search space.

2. *Ribbons share a common curve segment:*
   Ribbons that are formed by the same curve segment are likely to belong to the same object; and hence, they can be included in the same group.

3. *Ribbons are collinear with each other:*
   Object boundaries can be broken due to noise or other inadequacies of the low-level processes. Even after joining them on the basis of continuity, fragmentation may still persist. However, as indicated by the DNA model, fragmented ribbons of a linear DNA (as opposed to a curved one) should still be collinear, i.e., they are mutually supportive and can be grouped together.

4. *Ribbons have uniform width:*
   In general, the variation in the width of a DNA is small. If a particular symmetry has a width

that is significantly different from other ribbons, then we conclude that it is not in the same group.

These geometric constraints, together with the object model and high level features, establish the basis for the grouping process.

3.2.3 Geometric grouping

This step of the computational process attempts to assign a tag (unique ID) to each DNA object. This is accomplished by utilizing the geometric constraints in a stepwise fashion. However, the labeling consistency constraint defined by the model may not be guaranteed in this process. Hence, a set of verification steps are required to ensure the labeling consistency.

1. *U-shape consistency verification:*
   Every DNA has only two U-shapes and these are located at its ends. If a set of curve segments form more than two U-shapes, then some of these U-shapes must be erroneous. We distinguish these false U-shapes by a width property that is inconsistent with the width of other U-shapes.

2. *Linking consistency checking:*
   The two U-shapes must be connected by anti-parallel lines. We call this the "link consistency constraint", which is used to remove false U-shapes that are generated by erroneous linking of edgels obtained from Canny operator.

The grouping strategy is very similar to [8], and more details can be found in our earlier paper [12].

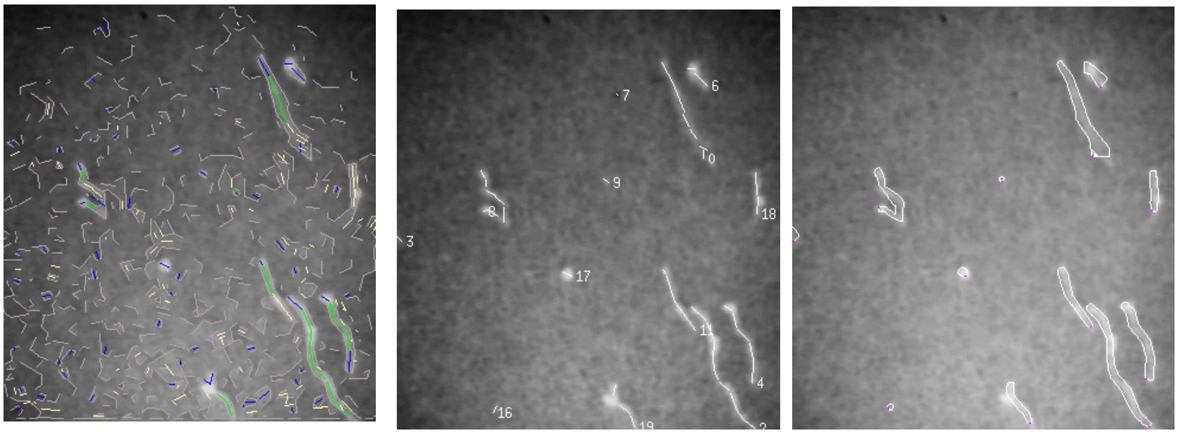
3.3 Example

We illustrate the result of our system on an image that is noisy and has a hot spot in its lower center of figure 11. In this example, the first image corresponds to edges and local symmetries; the second image corresponds to global symmetries; and the third image represents the inferred polygons. Note that for the curved molecule, there is no local symmetry where the molecule bends. In this case, a local symmetry is inferred to maintain continuity.

4 Tracking subsystem

The detection subsystem provides the initial contours in the form of bounding polygons for the first frame. This initial contour is then refined in the first frame, and projected onto the next frame so that tracking takes place. In this context, detection of tubular molecules occurs only in the first frame for the purpose of initialization, as shown in figure 9. The contour refinement algorithm [2] uses a cost function that is optimized by using dynamic programming. The cost function encodes the desirable properties of the refined contour in terms of high and low level feature activities. The low level features refer to pixel level information, such as local edge magnitude and direction.

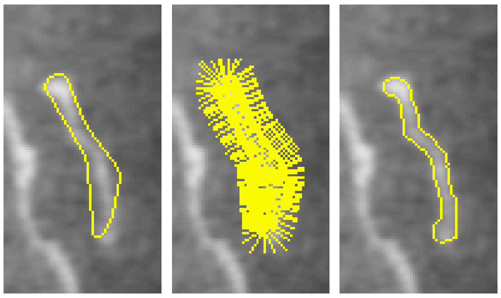
[2] In our system, the functionality of refinement and tracking is achieved with the same algorithm.



(a) (b) (c)

Figure 3: (a) Edges and local symmetries for a multiple molecules; (b) Global symmetries; (c) Inferred polygons.

In our system the high level constraints, derived from the bounding polygon, affect the contour refinement in two ways. The first is a geometric constraint, discussed later; the second limits the scope of the search. This is accomplished by smoothing the initial polygon with a Gaussian kernel and bounding the refined contour to lie in a small neighborhood as defined by the normal lines to the smooth curve. The rationale for Gaussian smoothing is that the bounded polygon, as shown in figure 11, is not smooth and the normal lines may not intersect the actual boundary of molecules. However, by smoothing the bounding polygon, the normal lines scan the molecule along its real boundary smoothly. The relationships between the smooth bounding polygon, normal lines, and the refined contours are shown in figure 12.



(a) (b) (c)

Figure 4: Steps in refinement for a single molecule: (a) Gaussian-smooth bounding polygon from the detection step; (b) Normal lines to the smooth polygon; (c) Refined polygon.

Conceptually, each point on the refined contour should have high gradient and good continuity with both the low and high level features. This is reflected in the cost function in terms of making a decision for a particular path. The cost function indicates the accumulation of making a decision to link two points from two consecutive normal line segments as shown in figure 13. Let:

1. g_i be the gradient magnitude for a point p_i on line L_i,
2. θ_i be the local edge direction at location p_i. We define the local edge direction as the direction that is normal to the direction of maximum rate of change of the local gradient,
3. $\Delta\theta_{ij}$ be the difference between the direction of gradient for points p_i and p_j,
4. γ_i be the angular difference between local edge direction and local direction of high level constraint. This is obtained by representing a local segment of the high level constraint, expressed by the smooth polygon, as a vector, and imposing that this vector should be *co-directional* with the local edge direction along the corresponding normal line.
5. η_{ij} be the angle between the local edge direction at point p_i and the vector connecting point p_i to p_j.
6. w_{ij} be the distance between points p_i and p_j located on two consecutive normal line segments, and
7. α_{ij} be the directional deviation between two consecutive points defined as $MAX\ (\Delta\theta_{ij}, \eta_{ij})$.

We formulate the cost function in such a way that the gradient is maximum, directional differences are small and the deviation for a particular path from the bounding polygon (high level constraint) is also small. In other words, the desired path should maximize the following cost function:

$$MAX_{i,j,k} \sum_k \sum_i \sum_j \frac{1}{w_{ij}} g_i e^{-tan(\frac{\gamma_i}{s})} \cos(\alpha_{ij}) \tag{3}$$

Where k is the length of the contour, i and j are the pixel locations on normal line to the smooth polygon, and s is a parameter that controls the amount of deviation from the high level contour (it is set to "2" in our program). This formulation indicates that: i) when $\gamma_i = \pi$,[3] the local cost function goes to zero, i.e., the local edge direction has an opposite direction to the direction of the smooth curve from the high level constraints, ii) if $\alpha_{ij} > \frac{\pi}{2}$, then the local cost will be negative and this particular sub-path will be inhibitive in the search process, and iii) by expressing the cost function as a product of internal and external energies, as opposed to the sum in equation (1), we have eliminated scaling coefficients that are used in the Snake formulation. The above cost function is optimized with the dynamic programming principle [15, 2, 12].

4.1 Examples

Two examples are given in this section. The first one shows the result of tracking of a single molecule over 140 frames. This molecule is unhooking at a pivot point and its mechanical properties are of significant interest to structural biologists. Six frames of this sequence are shown in figure 14. Note that the curved molecule stretches, and eventually becomes linear under electric field forces. The

[3] $0 \leq \gamma \leq \pi$

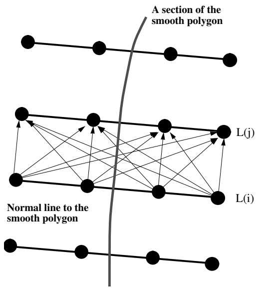
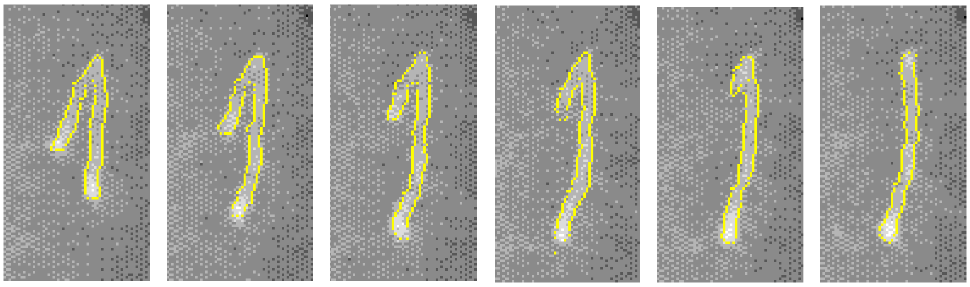


Figure 5: Possible paths for each point on a normal line segment



(a) (b) (c) (d) (e) (f)

Figure 6: Tracking a single molecule over 140 frames

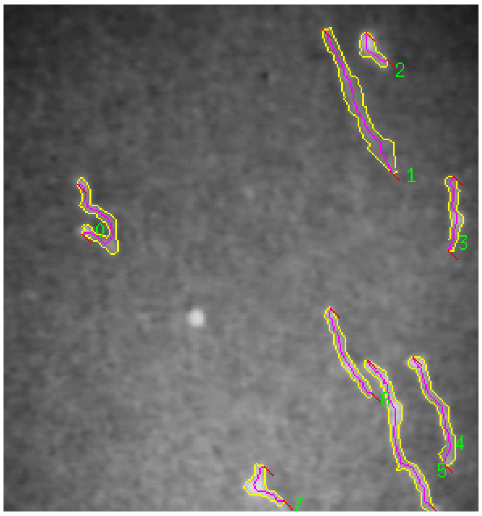
second example, in figure 7, shows tracking of several molecules over four frames of a video sequence, where each of these frames are roughly 120ms apart in time. These molecules are deformed as they are perturbed by an electric field. Note that some of the molecules lie very near each other, yet the tracking subsystem continues to track them without any interference.

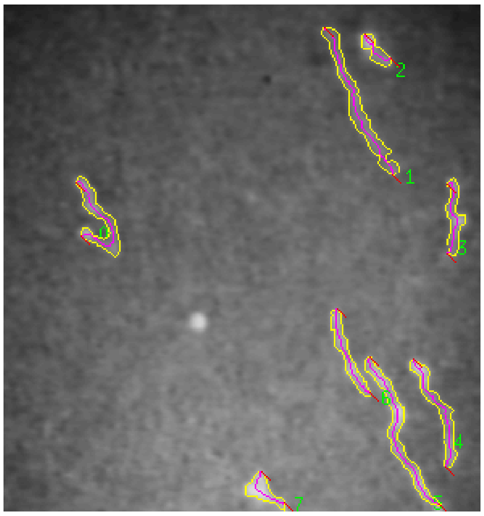
5 Conclusion

In this paper, we have shown a system for the detection and tracking of tubular molecules. We now summarize the limitations in the performance of our system, with special emphasis on the tracking algorithm.

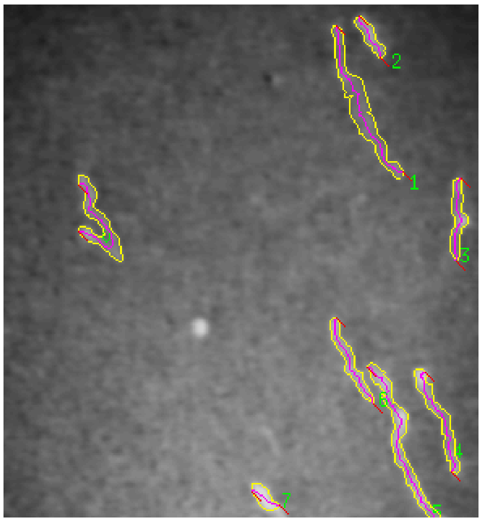
The most crucial parameters in the detection subsystem are the thresholds that are used prior to thinning in the Canny edge detector. If these thresholds[4] are set too high, then the result will have many missing edges that could lead to fragmentation in the grouping process. In other words, we assume that the local symmetries are correctly computed. In this sense, the system is unable to probe the image further and infer additional missing local symmetries. In our application domain, we have certain knowledge about the image history and the parameters associated with the imaging modality. This knowledge is used to set the corresponding parameters in the Canny edge detection

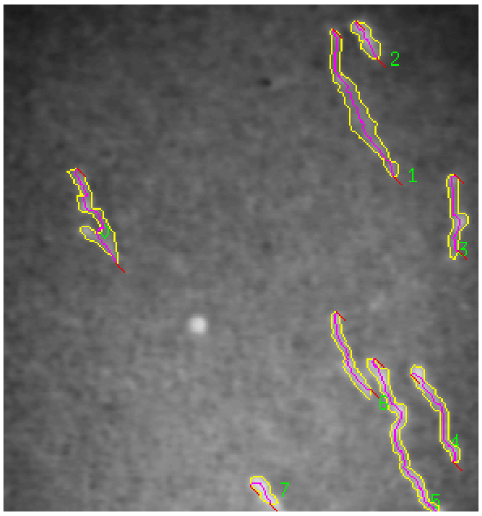
[4] Canny uses soft thresholding by means of a high and low threshold.





(a) (b)



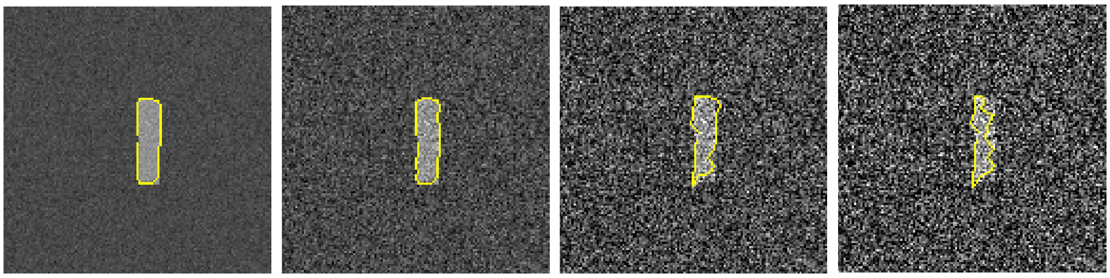


(c) (d)

Figure 7: Four frames of the tracking sequence with multiple molecules

for desirable performance. We have performed sensitivity analysis on these parameters by as much as 30%; no changes in the grouping have been observed.

The results of the tracking subsystem were shown on several hundred frames in the previous section on a single molecule as and with multiple molecules. The technique assumes that the normal lines to the smooth polygon intersect the actual boundary of the molecule. If the object deforms rapidly, then the tracking subsystem will not compute the correct boundary along the length of the molecule. From a qualitative perspective, the refinement process has a tendency to round sharp corners. Nevertheless, this is a characteristic of most snake-based techniques since they impose some form of the smoothness constraints on the results. From a quantitative perspective, we have tested the performance of the refinement algorithm under controlled environment with ground-truth information. This is accomplished by creating a synthetic thin bar image (a suitable model for a tubular molecules), adding Gaussian noise at different standard deviation, and measuring the error in the refinement. This is shown in figure 15. The refinement error is measured as a function of total angular changes along the contour. We represent each pair of adjacent points as a vector, and then accumulate the angle between adjacent vectors over the entire contour. This sum should be equal to 2π, and any deviation from it specifies the refinement error in radian.[5] In figure 15, we



(a) (b) (c) (d)

Figure 8: Errors in the refinement algorithm over a synthetic image with varying Gaussian noise. The background and the thin bar intensities are 80 and 140 respectively. The refinement error is expressed in radian. (a) Standard deviation of noise=10, error= 0.8; (b) Standard deviation of noise=25, error= 4.1; (c) Standard deviation of noise=47, error= 8.6; and (d) Standard deviation of noise=60, error= 13.4.

manually created an initial bounding polygon on the image and then used the prescribed algorithm for refinement. The error in the refinement is correlated with noise, and it is mainly due to abrupt changes in the value of edge gradient. In the second experiment, we created 10 different initial contours, and measured the mean and standard deviation in the refinement algorithm as a function of the noise. The results are tabulated in the following table.

S.D. of the noise in the image	10	25	47	60
mean of the refinement error	1.2	5.7	13.5	22.5
S.D. of the refinement error	.5	2.5	15.6	20

[5] We also looked at measuring the proximity of the points on the contour to their ideal position. However, we believe that the angular difference encodes error more effectively, since the length of the contour remains approximately the same.

The analysis reveals the effect of noise on the refined contour in terms of mean and standard deviation. It is clear that as the amount of noise is increased, the position of the refined contour becomes more unpredictable as a function of the placement of the initial contour. This was an expected result, which has been quantified in our studies.

Acknowledgement: The authors thank B. Crowley and N. Ostrofe for the preparation of this manuscript.

References

[1] A. Amini and etal. Mr physics-based snake tracking and dense deformations of tagged mr cardiac images. In *AAAI Conf. on Applications of Computer Vision to Medical Image Processing*, pages 126–130, 1994.

[2] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 855–867, 1990.

[3] R. A. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.

[4] C. Bustamante. Direct observation and manipulation of single dna molecules using fluorescence microscopy. *Annual Review of Biophysics and Biophysical Chemistry*, 20:415–446, 1991.

[5] C. Bustamante, T. Houseal, D. Beach, and F. M. Maestre. Fluorescence microscopy of the dynamics of supercoiling, folding, and condensation of bacterial chromosomes, induced by acridine orange. *Journal of Biomolecular structure and Dynamics*, 8:643–655, 1990.

[6] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[7] Q. Huang and C. Stockman. Generalized tube model: Recognizing 3d elongated objects from 2d intensity images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 104–109, New York, June 1993.

[8] A. Huertas, W. Cole, and R. Nevatia. Detecting runways in complex airport scenes. *Computer Vision, Graphics, and Image Processing*, 51(2):107–145, August 1990.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[10] F. Leymarie and M.D. Levin. Tracking deformable objects in the plane using an active contour model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:617–635, 1993.

[11] C. Nastar and N. Ayache. Classification of nonrigid motion in 3d images using physics-based vibration analysis. In *IEEE Workshop on Biomedical Image Analysis*, pages 61–69, 1994.

[12] B. Parvin, C. Peng, W. Johnston, and M. Maestre. Tracking of tubular objects for scientific applications. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 295–301, 1994.

[13] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:730–742, 1991.

[14] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:715–729, 1991.

[15] Bellman R. *Dynamic Programming.* Princeton University Press, 1957.

[16] L. Song and M. Maestre. Unhooking dynamics of u-shaped dna molecule undergoing gel electrophoresis. *Journal of Biomolecular structure and Dynamics*, 9:87–99, 1991.

[17] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *Computer Vision, Graphics, and Image Processing*, 55(1):14–26, 1992.