



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# Recovery and Visualization of 3D Structure of Chromosomes from Tomographic Reconstruction Images

Sabarish Babu, Pao-Chuan Liao, Min C. Shin,  
Leonid V. Tsap

April 30, 2004

IEEE Workshop on Articulated and Nonrigid  
Washington, DC, United States  
June 27, 2004 through June 27, 2004

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

# **Recovery and Visualization of 3D Structure of Chromosomes from Tomographic Reconstruction Images**

Sabarish Babu, Pao-Chuan Liao, Min C. Shin

Department of Computer Science  
University of North Carolina at Charlotte  
9201 University City Blvd  
Charlotte, NC 28223  
sbabu, pliao, mcschin@uncc.edu

Leonid V. Tsap

Advanced Communications and Signal Processing Group  
Electronics Engineering Department  
University of California Lawrence Livermore National Lab  
Livermore, CA 94551  
tsap@llnl.gov

## **ABSTRACT**

*The objectives of this work include automatic recovery and visualization of a 3D chromosome structure from a sequence of 2D tomographic reconstruction images taken through the nucleus of a cell. Structure is very important for biologists as it affects chromosome functions, behavior of the cell and its state. Chromosome analysis is significant in the detection of diseases and in monitoring environmental gene mutations. The algorithm incorporates thresholding based on a histogram analysis with a polyline splitting algorithm, contour extraction via active contours, and detection of the 3D chromosome structure by establishing corresponding regions throughout the slices. Visualization using point cloud meshing generates a 3D surface. The 3D triangular mesh of the chromosomes provides surface detail and allows a user to interactively analyze chromosomes using visualization software.*

**Keywords:** 3D Chromosome Structure, Region Segmentation, Medical Image Visualization, Medical Image Processing

# 1 Introduction

## 1.1 Motivation

Tracking and visualizing chromosomes gives biologists valuable information regarding their three-dimensional (3D) structure and behavior. Previously, segmentation of banded chromosomes frozen in metaphase of mitosis was important for classification especially in the karyotyping process. This process facilitates the classification and detection of chromosomal abnormalities such as Klinefelter's, Down's, and Turner's syndrome. Chromosome analysis is important in such situations as prenatal amniocentesis examination, detection of malignant diseases, and monitoring environmental gene mutations.

In this paper, we propose a new method for (1) an automatic recovery of chromosomes in a sequence of 2D images, and (2) visualizing the resulting chromosomes in 3D. Such 3D visualization provides biologists with information that cannot be obtained by 2D images alone. Various imaging systems rely on reconstruction of an image from its projections through the process of computed tomography (CT). In medical imaging, for example, X-ray plates, CT scans, magnetic resonance imaging (MRI), and various types of positron emission tomography (PET) all record 2D projections of 3D objects [14]. Hence, tracking the contour of an object along each successive slice allows us to recreate a 3D representation of the object. We see that 3D visualization of the chromosome can be useful for biologists in the following ways: (1) identifying the space occupied by the chromosome within the cell, (2) visualizing specific structures along the contour such as "constrict points," and binding sites with other intercellular molecules such as proteins, enzymes, and other organelles, (3) using the visualization to accurately classify the chromosomes, (4) detecting anomalies, such as chromosomal disorders,

and (5) helping to identify the behavior of the organelle over time (sometimes called 4D reconstruction, with time as the the fourth dimension).

## **1.2 Previous Work**

Previous research of chromosomes in 2D images was primarily focused on abnormality detection and classification of chromosomes. In chromosome classification (Karyotyping), one of the main efforts includes the problem of separation of partially occluded chromosomes. Lerner et al. [1] proposed classification based on skeleton points, and local feature extraction for classification purposes (CPOOS – Classification-Driven Partially Occluded Object Segmentation method). Shi et al. also used local features such as cut points, skeleton points, junction points, and ravine points to separate touching chromosomes using Parallel Mesh algorithm [3]. Some of the local features extracted were based on topology, such as concavities, that indicated where occlusions between chromosomes occurred. Hence, their separation and classification were based on landmarks and occlusion points. Lerner et al. [22] trained Multilayer Perceptron (MLP) Neural Networks to classify chromosomes and used a “knock out” technique as well as Principle Component Analysis (PCA) for feature selection. Both techniques yielded the benefit of using only about 70% of the available features to get the most out of classifier performance. Vidal et al. used syntactic/structural pattern recognition algorithms such as Error-Correcting Grammatical Interface (ECGI) and MLP to classify chromosomes by formulating rule-based string representation of the features extracted [2]. Keller et al. presented a fuzzy logic system in addition to neural network based classification system to deal with ambiguities during the classification process. These ambiguities included imprecisions in computation, and in-class definitions in mid-level computer vision processes [23]. Minor chromosomal abnormalities cannot be detected by applying available techniques to 2D images. Based on banding patterns

and skeletal line lengths, only major chromosomal abnormalities, such as deletions, can be detected. Inversion abnormalities may cause problems in information extraction for the formulation of rules in the syntactic/structural methods. However, 3D visualization enables scientists to discern occluding chromosomes for further classification better than 2D image analysis.

There have also been several papers published on visualization and 3D reconstruction of large and small biological objects based on various imaging modalities. The Visible Human Project of the National Library of Medicine used transverse CT, MR, and cryosection images of representative male and female cadavers to obtain 3D human body representations [20]. The cryosection images, used for full body visualization were taken at regular intervals, the male was sectioned at one-millimeter intervals, and the female at one-third of a millimeter intervals. Subramanian et al. used Intravascular Ultrasound (IVUS) images, which is a technology for imaging the vascular lumen and atherosclerotic plaque structure, and devised a technique to accurately reconstruct 3D geometry of blood vessels [21]. Various imaging modalities were also employed in 3D reconstruction such as biplane X-ray fluoroscopy, X-ray, and echo images. The path of the catheter tip was estimated by fitting an interpolating spline through the 3D points. Arnison et al. presented a modality called Differential Interference Contrast (DIC) microscopy and applied Hilbert transforms to distinguish features of chromosomes from background in each 2D slice, and selective opacity to 3D pixels (voxels) according to their intensity to visualize chromosomes [24]. Engelhardt et al. visualized metaphase chromosome from human (HeLa) cell lines using electron microscopy (EM) [25]. The images were aligned using colloidal gold particles as reference points, and reconstruction was produced by the weighted back-projection

method. Our work focuses on automatic recovery and visualization of chromosomes in tomographic reconstruction volume slices.

## **2 Methodology**

### **2.1 Overview of the Approach**

The objective of our research is to track the contour of chromosomes in a sequence of tomographic reconstruction images, thus enabling us to recover the chromosome object and to provide visualization. The dataset was generated in the Sedat Lab at the University of California San Francisco. The slices are grayscale images of two chromosomes of the common fruit fly (*Drosophila melanogaster*). The slices progress along a plane of capture, and total sixty-five slice images.

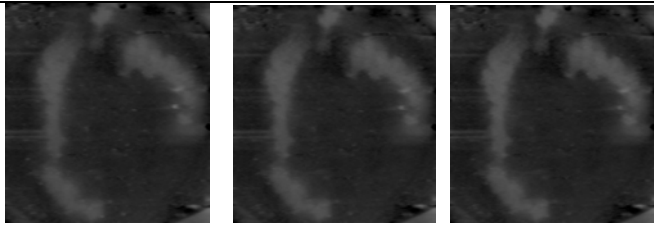



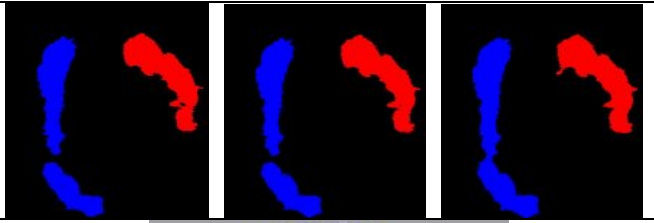
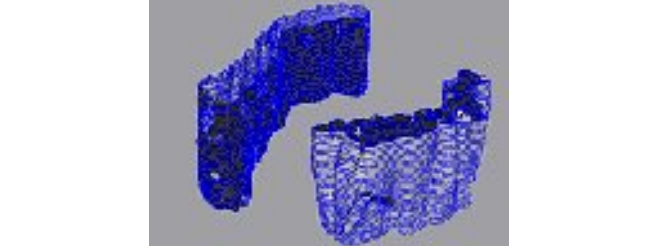
Our proposed methodology consists of five stages.

1. Segmentation of the chromosome regions in each 2D image slice is performed by image thresholding. The threshold is automatically selected by analyzing the histogram contour using a polyline splitting algorithm [7].
2. Noise removal is achieved by connected component labeling (CCL) [7] to filter out foreground regions below a certain size.
3. Two-dimensional contour refinement on each slice is performed on the contour of the chromosome regions extracted after step 2. This step employs an active contour model (snake) technique [9].
4. Region correspondence is performed by tracking the 2D regions of the same chromosome in adjacent slices. We use a region comparison method proposed by Hoover et al. [8] to correspond regions of the same chromosome between slices. This method achieves tracking even when the chromosome breaks into multiple regions in some slices.



5. Visualization of the chromosome in 3D consists of two steps. Initially, we extract a set of nodes from the contour of a single chromosome in each slice using chain-coding algorithm [7]. This set is taken for each chromosome, to create point clouds. Then, using meshing technique [13], we construct a mesh representing the surface of each chromosome.

These steps are described in separate sections in this paper. Figure 1 illustrates the flow chart of our methodology, showing sample output images from each step described above.

<p><b><u>Input Images</u></b>  The three sample input images on the right are from the set of tomographic reconstruction volume images. The total number of slices is 65, and is greyscale. From left to right, the image samples are slice number 23, 24, and 25, respectively.</p>	
<p><b><u>1. Region Segmentation</u></b>  Sample binary images from left to right are output images after region segmentation has been performed.</p>	
<p><b><u>2. Noise Removal</u></b>  Images on the right show the result of noise removal performed on the output images of region segmentation.</p>	
<p><b><u>3. 2D Contour Refinement</u></b>  Using the resulting contour as an initial estimate, we apply active contour models (snakes) [9] to refine it. Image on the right shows the refined contour of chromosome 1 (red) in slices 23, 24, and 25, respectively.</p>	
<p><b><u>4. Region Correspondence</u></b>  Tracking these chromosomes in adjacent slices (region correspondence) yields blue regions corresponding to chromosome 1 and red regions corresponding to chromosome 2.</p>	
<p><b><u>5. Visualization</u></b>  Points along the refined contour corresponding to each chromosome (nodes) are extracted to obtain point clouds. Meshing technique [13] is applied to the point clouds to obtain the mesh describing the surface of the chromosome.</p>	

**Figure 1:** Flow chart of the methodology showing sample input images, and output images corresponding to steps outline in the left column.

## **2.2 Dataset**

Various imaging systems rely on reconstruction of an image from its projections through the process of computed tomography (CT). In medical imaging, for example, X-ray plates, CT scans, magnetic resonance imaging (MRI), and various types of positron emission tomography (PET) all record 2D projections of 3D objects [14]. Our dataset consists of tomographic reconstruction of 2D chromosome image slices. There are a total of 65 images representing slices taken along a plane of capture. Most medical imaging systems separately reconstruct 2D slices of a 3D object. Those slices closest to Slice 1 are more indistinct, consisting primarily of background. Some of the slices, in particular ones closer to the end of the sequence, contain significant noise, which makes the task of chromosome segmentation more difficult.

## **3 Region Segmentation and Noise Removal**

### **3.1 Overview**

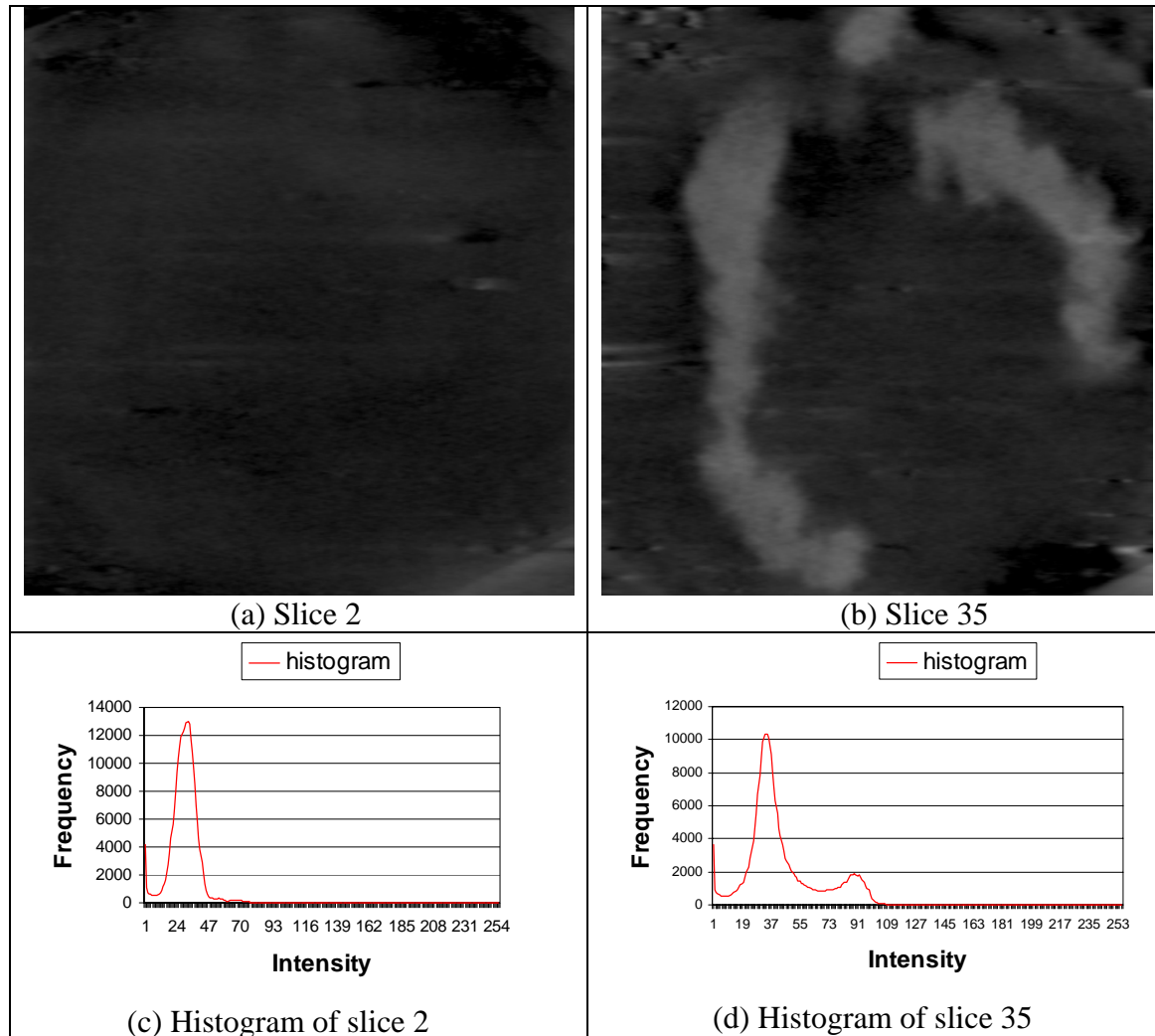
The goal of this step is to create a methodology that allows us to segment chromosome region from its background in each 2D slice, through automatic selection of an appropriate threshold for segmentation. In this step we also remove slices for which suitable thresholds could not be determined. The histograms of the slices are either unimodal or multimodal. The histograms of the end slices are unimodal, as the end slices contain only background. The histograms of the middle slices, however, are multimodal. To perform thresholding and subsequently segment chromosomes from the background successfully, we must establish that the histograms of the slices are multimodal.

Segmentation of many clinical and biological images is currently performed using manual slice editing. [6]. This method has some deficiencies, such as difficulty in achieving reproducible results, operator bias, and it is tedious to perform. Segmentation using techniques

such as region growing, edge detection, and mathematical morphology operations, mostly requires considerable amounts of expert interactive guidance because some knowledge of the domain (the content of images) is necessary. Hence, automatic segmentation with little to no human intervention would be preferred.

Applying polyline splitting on the contour of the histogram enables us to determine whether the histogram is unimodal or multimodal, subsequently excluding slices with a unimodal histogram. This method also provides a basis for threshold selection through the peakiness method [7], which will be described in the following section. Details of the polyline splitting algorithm can be found in [7]. By using the polyline splitting algorithm on a histogram contour, we can find the list of edges with vertices end-to-end that describes the histogram curve by recursively splitting it into line segments.

Otsu's method of thresholding (recommended by Shi et al. [3]), based on minimizing intra-group variance and maximizing inter-group variance, is not applicable to this dataset. These algorithms assume that the histogram is bimodal and demonstrates essentially two distributions. Hence, we seek an automatic method that analyses the histogram contour of each slice to determine whether it is unimodal or multimodal, excludes unimodal slices, and selects an appropriate threshold based on a reasonable measure such as the peakiness method. Slice 2 (Figure 2a), one of the top end slices, is unimodal (Figure 2c) and it is therefore very difficult to distinguish foreground from background. In contrast, slice 35 (Figure 2b), one of the images from the middle slices, is bimodal (Figure 2d). Hence, it is possible to extract the chromosome from background using region segmentation.



**Figure 2:** (a) Slice 2 is one of the earlier slices of the CT scan images with a unimodal histogram distribution of intensity. (c) The histogram for slice 2. (b) Slice 35 is one of the middle slices with a bimodal histogram distribution of intensity suitable for thresholding. (d) The histogram for slice 35.

### 3.2 Description of the process

The goal of this step is to robustly segment chromosome regions by (1) determining whether the image contains any chromosome and (2) finding the correct threshold even when the histogram of image contains multiple modalities. The polyline splitting algorithm is used to analyze the contour of the histogram [7]. It iteratively divides a curve into a set of line segments denoted by a set of vertices (see Figure 3). If we detect no local minima in an image slice, we

determine that the histogram of the image is unimodal; thus the image does not contain any chromosomes (see Figure 3a.) When a local minima is detected, we find the threshold by finding the local minima ( $k$ ) with the highest peakiness [7]. The peakiness as

$$\min\{H(i), H(j)\} / H(k) \quad (3.1)$$

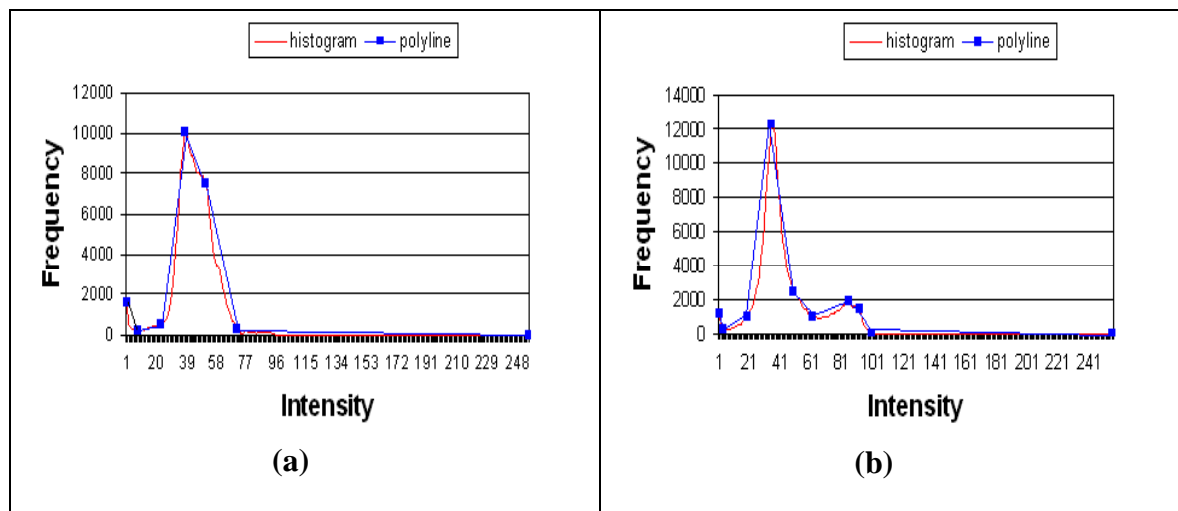
where  $i$  and  $j$  are the intensity value of the neighboring local maximas and  $H(x)$  is the histogram value at the intensity of  $x$ .

### 3.3 Results of Region Segmentation

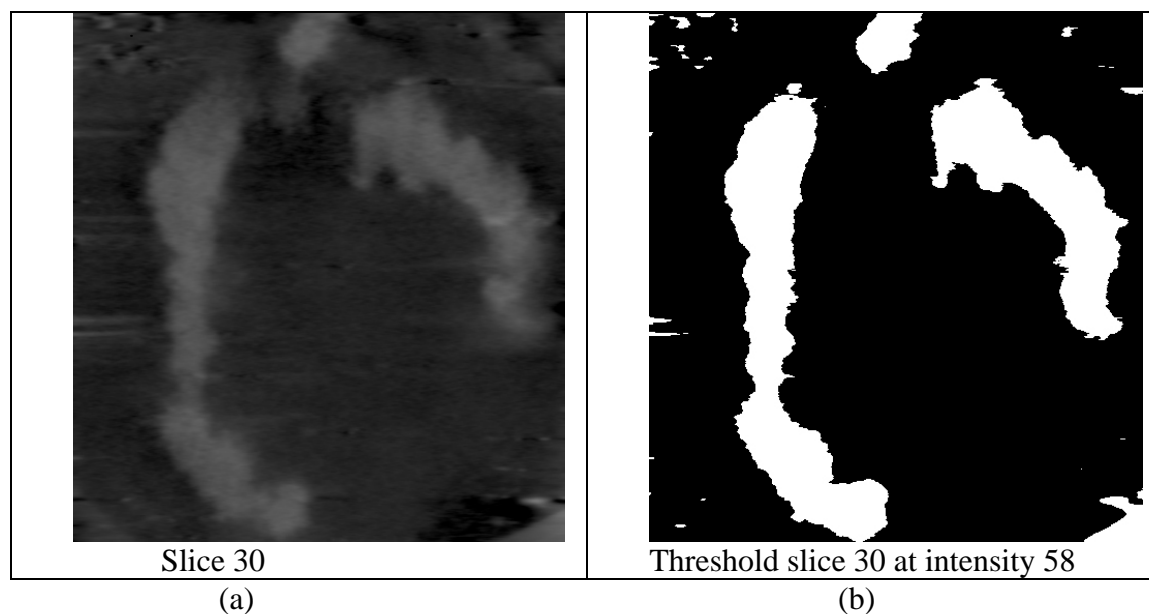
In Figure 3, no local minima is present on the histogram for slice 9 (Figure 3a), while one local maxima is found at intensity 37, thus indicating that the histogram for slice 9 is unimodal. After applying polyline splitting to the histogram contour of the entire sequence, slices with unimodal histogram are excluded from the subsequent processes performed for visualization. They include slices 1 through 6, 9, 13 through 22, and 54 through 65.

Figure 3b shows local maxima 1 and local maxima 2 found with a local minima containing the highest peakiness value in-between these two local maxima. The intensity of this local minima is selected, and this value is used for region segmentation throughout the process of thresholding. A polyline distance threshold value of 25 enables proper distribution of vertices in the polyline, giving rise to a single minima vertex placed in-between two local maxima vertices.

Figure 4 represents the results of polyline splitting of the histogram contour, applied to the original image (Figure 4a) with the threshold point at intensity 58. Thresholding then extracts the foreground chromosome objects (white represents intensity 255) from the image, and the rest is labeled as background (black is intensity 0). The table in Figure 5 shows results for every fifth slice followed by threshold values and peakiness estimates.



**Figure 3:** (a) Histogram for slice 9 with polyline (blue) modeling the contour of the unimodal histogram curve (red), (b) Histogram for slice 30 with polyline (blue) modeling the histogram contour (red) of the bimodal histogram curve.



**Figure 4:** (a) Slice 30 in its original form, and (b) binary thresholded image transformation of slice 30 to reveal white (foreground) chromosome regions, and black background.

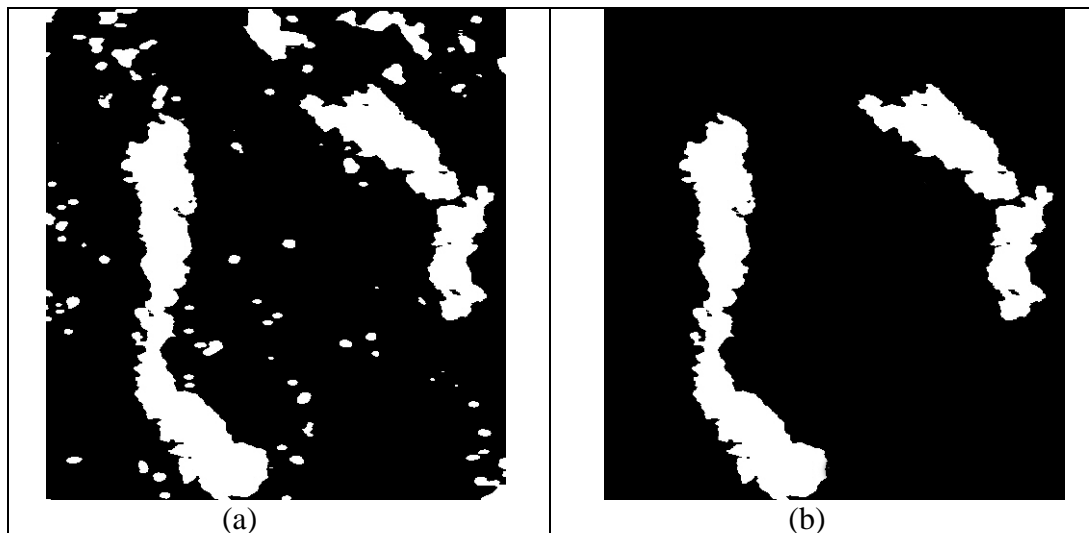
<u>SLICE</u>	<u>PEAKINESS</u>	<u>THRESHOLD</u>
1	0.0	Not Found
5	0.0	Not Found
10	1.614	73
15	0.0	Not Found
20	0.0	Not Found
25	1.358	59
30	1.762	58

35	2.054	62
40	2.005	58
45	1.764	59
50	1.527	64
55	0.0	Not Found
60	0.0	Not Found
65	0.0	Not Found

**Figure 5.** The computed threshold for every fifth slice is shown. Note that a threshold is not found for slices with the maximum peakiness of 0, indicating that those histograms are unimodal.

### 3.4 Noise Removal

After region segmentation, small region removal is performed to eliminate noise in the thresholded image. Regions are detected by using connected component labeling [7] and the regions smaller than 500 pixels are removed.



**Figure 6:** (a) Slice 53 after region segmentation and (b) after small region removal.

## 4 2D Contour Refinement

### 4.1 Overview

The objective of this step is to refine the contour of the chromosome regions in each 2D slice of the volume data, prior to region correspondence, so that we may obtain an accurate contour of the chromosome regions in each 2D slice. The process employs 2D snakes to refine



the contour in each slice. A snake, also known as an active contour model, builds a controlled continuity spline governed by an energy function under the influence of image and external constraint forces. The snake can be either closed or open. To obtain a desired contour, the snake is moved to a minimal energy condition. The total energy can be described as

$$E = \int_0^1 E_{\text{int}}(v(s)) + E_{\text{image}}(v(s)) + E_{\text{con}}(v(s)) ds \quad (4.1)$$

where  $v(s) = (x(s), y(s))$ ,  $E_{\text{int}}$  indicates the internal force of the spline due to bending,  $E_{\text{image}}$  denotes the image influence, and  $E_{\text{con}}$  represents the external constraint forces which permit to control a snake interactively.

Several different energy functions have been developed, as well as the calculation of each force term. In this paper, snake implementation is based on [10] due to speed and stability considerations. A brief explanation of this method is included below.

## 4.2 Internal Force

To describe the bending effect, two terms are considered:

$$E_{\text{int}} = (\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2) / 2 \quad (4.2)$$

The first-order term controlled by  $\alpha(s)$  causes the snake to act like a membrane, and the second-order term controlled by  $\beta(s)$  makes it act like a thin plate. This modifies the behavior of a snake to be more membrane-like or thin-plate-like by adjusting these two coefficients. Notice that when  $\beta(s) = 0$ , it represents a corner.

If  $v_i = (x_i, y_i)$  is a node of snake, then the first-order term can be calculated easily by a backward finite difference approximation:

$$\left| \frac{d^2 v_i}{ds^2} \right|^2 \approx |v_i - v_{i-1}|^2 = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \quad (4.3)$$

The second-order term represents the curvature in  $v_i(x_i, y_i)$ , which can be computed as:

$$\left| \frac{d^2 v_i}{ds^2} \right|^2 \approx (\Delta x_i / \Delta s_i - \Delta x_{i+1} / \Delta s_{i+1})^2 + (\Delta y_i / \Delta s_i - \Delta y_{i+1} / \Delta s_{i+1})^2 \quad (4.4)$$

$$\text{where } \Delta x_i = x_i - x_{i-1}, \Delta s_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (4.5)$$

More details and discussion can be found in [10].

### 4.3 Image Force

This term can provide a force that attracts the snake toward the desired feature of the object. In [9], the authors present three different force functions, those attracting a snake to lines, edges, and terminations. In this research, edge function is chosen to be the image force function.

The energy function becomes:

$$E = \int (\alpha(s)E_{cont} + \beta(s)E_{curv} + \gamma(s)E_{image})ds \quad (4.6)$$

where  $E_{cont}$  and  $E_{curv}$  are the first- and second-order terms of internal forces described in 4.1, respectively. As previously, one can adjust the weight of this term by changing the value of  $\gamma$ .

### 4.4 Greedy Algorithm

Once a snake and its energy function are decided, a greedy algorithm is used to find the best contour that has minimal energy. A greedy algorithm has been proved faster than dynamic programming and also more stable and flexible than variational calculus.

The algorithm can be outlined as follows:

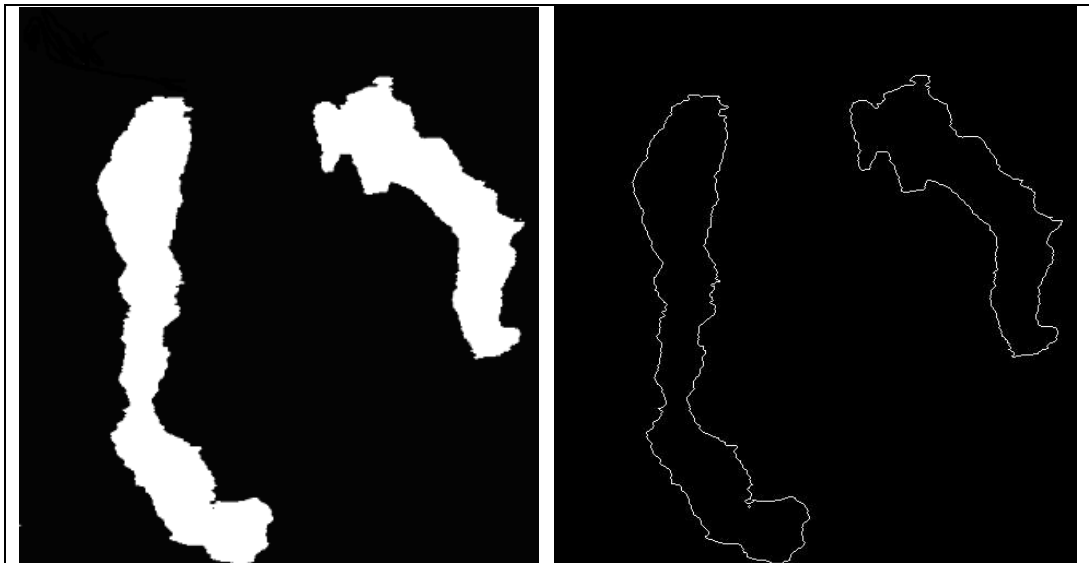
- A. For each node of the snake, evaluate the total energy at its neighbor point (8 points) then move each node to its neighbor point that has minimal energy.
- B. After finding the minimal energy for all nodes, corners are checked. If a node satisfies conditions below, it could be a corner, which sets  $\beta$  to 0. This means that the curvature constraint is off.

- a. Curvature is larger than for neighbor nodes.
- b. Curvature is larger than threshold.
- c. Edge strength is above threshold.

Repeat steps a and b until the snake does not move anymore, or the number of moved nodes is below a threshold. In this research, if the moved node is less than 5% of total nodes or the iteration is more than 1000, we consider that the snake is in stable condition and has converged.

#### 4.5 Contour Extraction from Images

The initial contour of a snake is estimated by performing chain coding [7] after noise removal. An example is shown in Figure 7.



**Figure 7:** A slice before and after contour extraction.

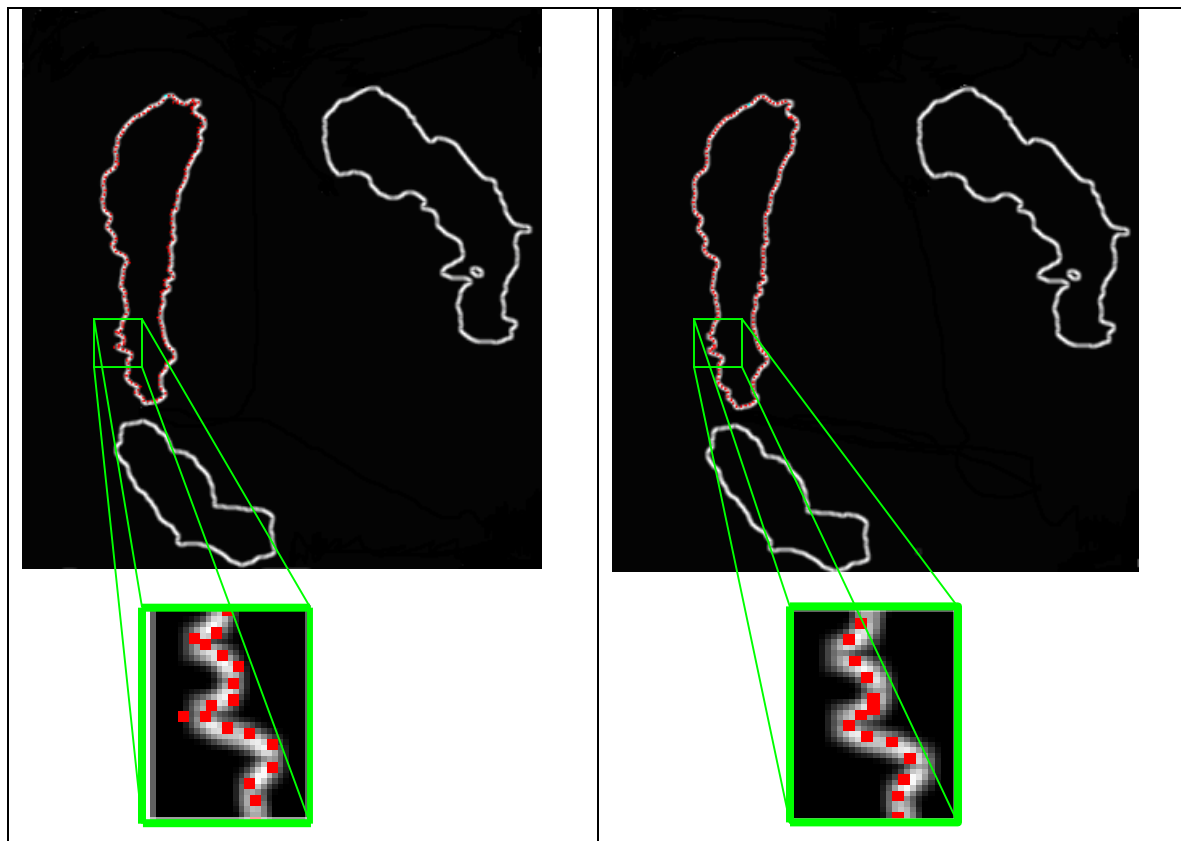
#### 4.6 Preprocessing before Snake Application

Before a snake can work on the images, additional image processing techniques must be performed on the original grayscale images so that the image forces can be computed correctly. First, a double-level thresholding with values  $T_{low}$  (lower threshold bound) and  $T_{high}$  (upper threshold bound) is applied. The original threshold value from region segmentation represents

an initial estimate for  $T_{low}$  and  $T_{high}$ . Any pixel with a value between  $T_{low}$  and  $T_{high}$  belongs to a foreground, while everything else constitutes background. Next, a median noise reduction is performed. A Gaussian smoothing algorithm is then employed to blur the image. Sobel edge detection is used to find the edges; and the resulting image is called a normal Gaussian image. Then Gaussian smoothing is performed again resulting in a smoothed Gaussian image. At the beginning, we use a smoothing Gaussian image to attract the snake. When the snake is close enough, normal Gaussian image is then used to revise the snake to the correct edges.

#### 4.7 Results of 2D contour refinement

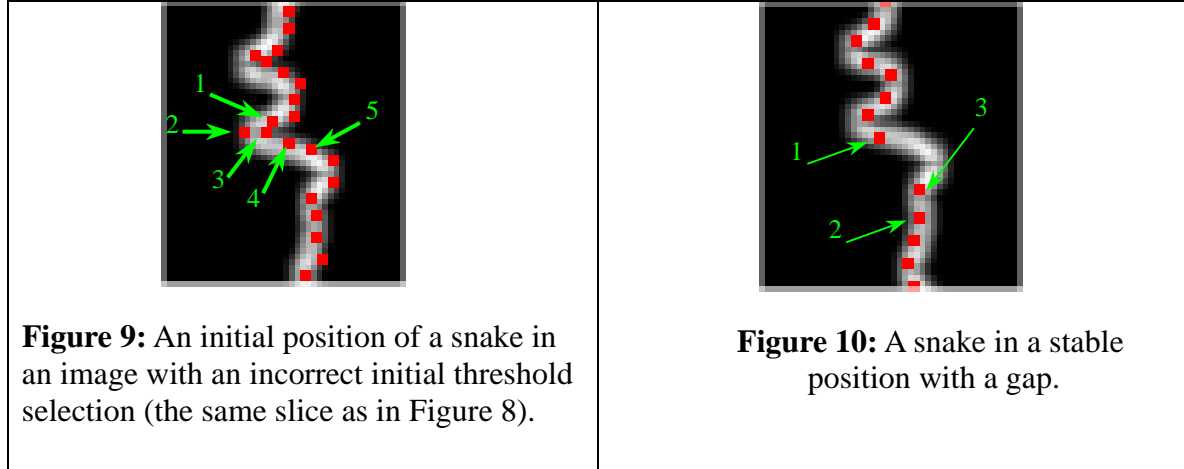
Since a single particular energy function is not suitable for every domain, combinations of the coefficients have been tested to achieve results. An empirically selected set of parameters produces the results shown in Figure 8. Through zooming (Figure 8), we can find that the snake nodes are attracted by the edges successfully, and a more accurate contour is obtained.



**Figure 8:** (Slice 23, Tlow = 61, Thigh = 110) Initial snake position (left), and snake in a stable position with minimal energy (right). Enlarged images show the detailed changes of the snake.

If the initial threshold estimate is incorrect, then the algorithm does not perform optimally, and improper node placement can result. Figure 9 is extracted from the same slice as the one in Figure 8, but with a lower Tlow at 60. The detected edges with lower Tlow have moved slightly away from the center when compared to Figure 8. The order of this snake is marked from 1 to 5. When the snake starts moving, the node marked 3 could move to a higher edge or a lower edge. If node 3 moves in-between 1 and 2, the order of the snake points becomes incorrect. Then, node 4 tends to follow node 3 due to the continuity force. Also, node 5 will follow node 4, and so on. The method used in this research to avoid such problems checks for nodes at conflicting positions. If so, one of the nodes will be deleted. For example, if node 3 moves to node 1, then node 3 will be deleted, and the order becomes 1, 2, 4, and 5. In the meantime, it is quite likely that node 4 will move to node 2's position because they have the same internal energy. If this does happen, node 4 is deleted as well. The order now becomes 1, 2, and 5. The order is now correct but the number of nodes has decreased. This may occur in various parts of a snake, which would become shorter and would re-form again trying to achieve an even point distribution.

One possible resulting problem is the occurrence of gaps, such as that between nodes 1 and 2, shown in Figure 10, once the snake has reached a stable state. The reason node 2 does not move closer to position 3 is as follows. For node 2, the energies in position 3 and its current position are almost the same. Therefore, a snake may be attracted to a very sharp point.

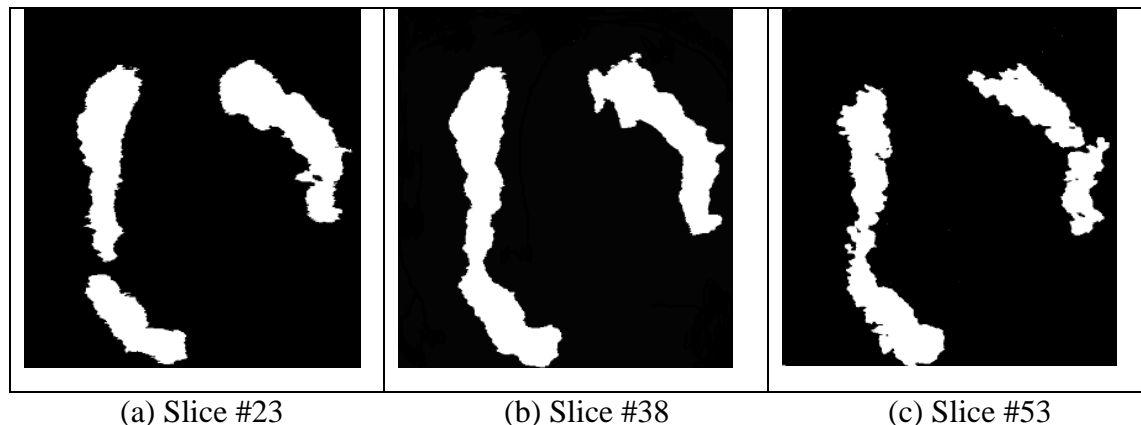


## 5 Region Correspondence

The objective of this step is to track 2D foreground regions of the same chromosome, which are corresponded by comparing adjacent slices, and to recover the set of regions comprising a 3D structure. During the previous step of 2D contour refinement, a more accurate contour of the chromosome regions in 2D is established after segmentation and noise removal. The 3D structure is produced by a set of 2D regions of same chromosome. We establish the correspondence between the slices by using the region comparison scheme proposed by Hoover et al. [8] that was used for range image segmentation comparison. Given a pair of images with segmented regions, the scheme classifies each region into five categories of correct, missed, noise, over- and under-segmentation. After the noise removal and 2D contour refinement, pairs of images are considered for regions correspondence learning. As the process iterates through the candidate pairs in the sequence, regions corresponding to 2D chromosome structure in each slice are tracked, producing sets of regions corresponding to a 3D chromosome.

Figure 11 represents two instances of segmented images from end slices (the left image is from slice 23, and the right one is from slice 53) with disjoint regions in chromosomes

(chromosomes 1 and 2, respectively). The image in the middle is slice 38 after segmentation, which is a mid-slice in the sequence with both chromosomes well-defined.



**Figure 11:** Corresponding regions among slices. Note that the two chromosomes shown as two regions in slice #38 could be broken up in slice #23 and slice #53.

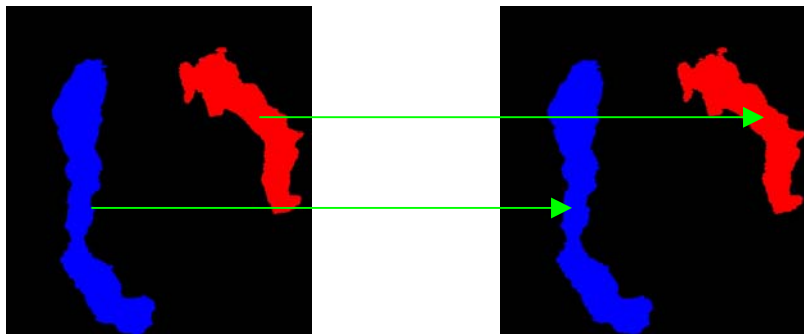
Among the five different types of classifications [8] namely correct, over-segmentation, under-segmentation, missed, and noise, we are interested primarily in the correct, over- and under- segmented regions, as it is highly unlikely that the tracked chromosome could be classified as missed or in noise regions. Metrics are designated as follows:

- i. An instance of **correct** classification is specified when a pair of regions in the adjacent images has at least  $T$  percent of the pixels in the chromosome region  $R_i$  in the first image marked as pixels in chromosome region  $R_j$  in second image. This follows from the premise that  $R_i \cap R_j \neq \theta$  for all  $i$  and  $j$ , if  $i = j$ .
- ii. An instance of **over-segmentation** classification is specified when a chromosome region  $R_i$  in the first image and a set of regions in the second image  $R_j^1$  to  $R_j^n$ , has at least  $T$  percent of the pixels in each chromosome region  $R_i$  in the first image marked as pixels in the union of chromosome regions  $R_j^n$  of the second image.

iii. An instance of **under-segmentation** classification is specified when a set of chromosome regions in the second image  $R_j^1$  to  $R_j^n$  and a chromosome region  $R_i$  in the first image has at least  $T$  percent of the pixels in chromosome region  $R_i$  in the first image marked as pixels in the union of chromosome regions  $R_j^1$  to  $R_j^n$  in the second image.

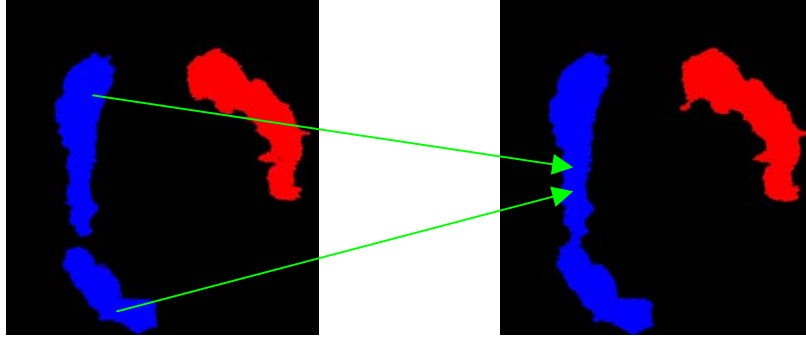
Applying the metric described above allows us to track chromosome regions among adjacent slices through the sequence of volume image slices and allows recovery of sets of chromosome regions that correspond to a 3D chromosome. Two resulting sets of regions corresponding to both chromosomes are shown in Figure 12 (blue and red). When an instance of over-segmentation is detected among the adjacent slices, we are able to track the disjoint regions of a chromosome in one image as belonging to a single region in a different image. This method allows us to identify disjoint regions in one image as belonging to a single region in another image by classifying the observed instance as under-segmentation.

This method is used to recover regions in images that correspond to chromosomes 1 and 2 labeled in blue and red, respectively. Figure 12 shows color images of slices representing examples of correct, over-, and under-segmentation.

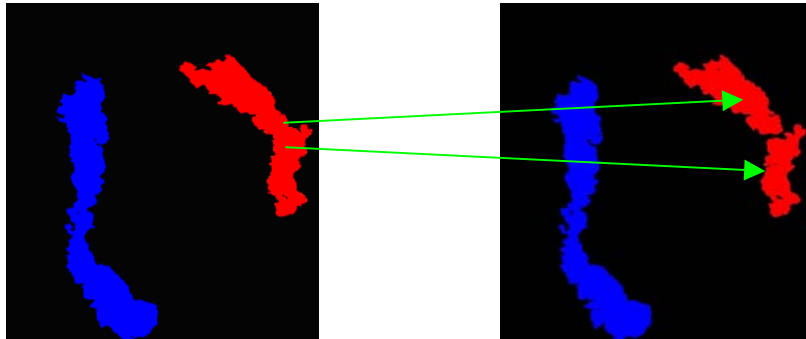


(a) Correct segmentation between slice #37 (left) and slice #38 (right).





(b) Under-segmentation of region of slice #24 (left) in slice #25 (right)



(c) Over-segmentation of region of slice #52 (left) in slice #53 (right)

**Figure 12:** Results depicting correct, over-, and under-segmentation respectively in each row. All of the instances have been reclassified as either chromosome 1 (blue) or chromosome 2 (red). Green arrows show the correspondence.

## 6 3D Visualization

### 6.1 Overview

The objective of this stage is to extract points corresponding to the contour of each chromosome object and visualize the point clouds in 3D. Such contour points are collected from each 2D slice to build a set of 3D point clouds. Subsequently, a 3D point-cloud-meshing method is applied to each set to reconstruct the surface mesh of each 3D chromosome.

### 6.2 Point Cloud

A point cloud is a set of points. The points can be 2D or 3D points and also can be categorized into unorganized point clouds or structured point clouds. An unorganized point cloud is a point set that has only spatial position and no other information such as geometry, or

shape. By contrast, a structured point cloud provides additional information that can be used for meshing, e.g., break-lines. In implementation, the algorithm dealing with unorganized point cloud usually transfers the data into structured data based on their coherence before generating the surface mesh. In this research, our data is actually a 2.5D point set, since the z-axis is the interval between two slices, which is a constant value. However, we treat the data as a 3D unorganized point cloud.

### **6.3 Meshing**

Meshing is the process of generating a consistent polygon model (mesh) from a given point set. The algorithm requires producing vertices, edges, and faces with shared vertices and edges. In many approaches, finite element technique is used to find the optimal mesh. We classify algorithms as 2D, 2.5D, or 3D, according to the data sets on which they operate. Usually quadrilateral or triangular meshes are generated in 2D and tetrahedral meshes are generated in 3D.

According to Remondino [13], the procedure for meshing can be divided into the four steps noted below.

*Preprocessing* – reduces computation time by sampling or eliminating points.

*Evaluation* – globally estimates the objects' surfaces and the relationship between adjacent points.

*Generation* – core step that generates vertices, edges and faces.

*Post-processing* – refines the mesh, such as filling holes.

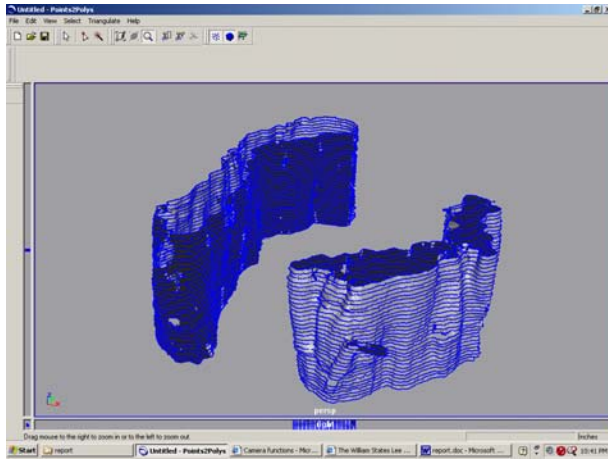
For detailed information on the meshing algorithm, we refer the reader to [13].

## 6.4 Results of 3D Visualization

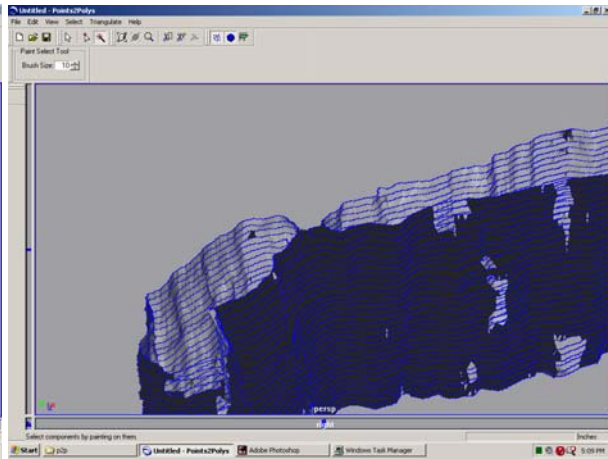
To visualize the 3D chromosome objects, Points2Polys is used together with OpenGL. This method takes point clouds as input and generates triangular meshes automatically. It also provides a function that optimizes the number of points, thus requiring fewer meshes and speeding up computation and visualization. Before we import the point cloud into this software, we code a simple program to combine all the slice nodes together, with a small z-interval value for distance between slices. An overview of the 3D chromosome model is shown in Figure 13. Figure 14 shows the point at which the chromosomes split up toward the end slices, which can be observed in the 3D model. Figure 15 shows a protrusion on the surface of the chromosome and the corresponding 2D image. This protrusion could be due to errors in segmentation or could indicate binding between an intracellular object (such as proteins or mRNA) and the chromosome.

An inconsistent surface generated from a missed initial guess for the 2D contour during the refinement process results in a hole in the 3D model. These holes can be filled by changing parameters in the meshing algorithm to accommodate distant vertices in the meshing process.

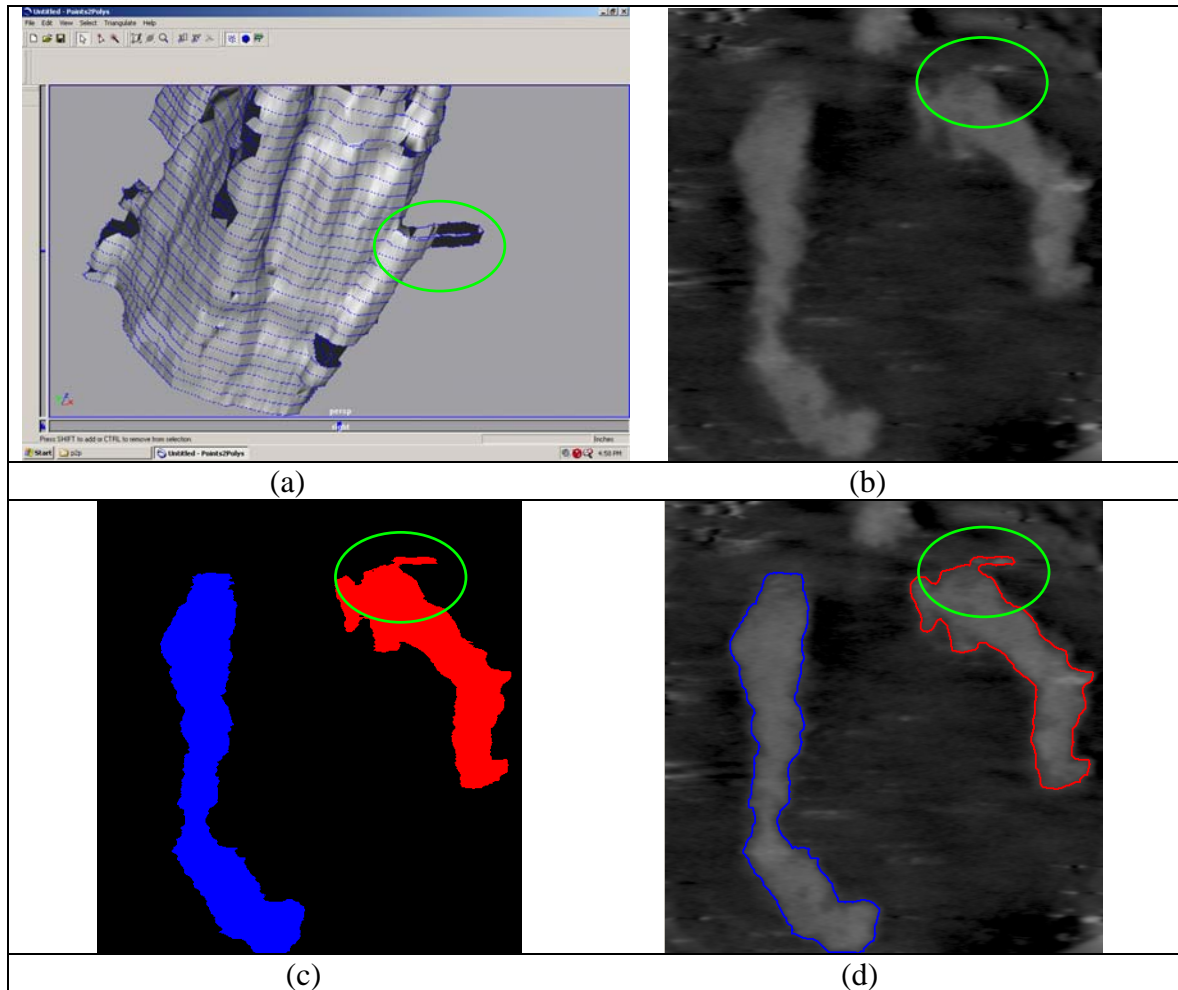
From Figures 13, 14, and 15a, one can recognize the relationship between each 2D cross-section through 3D visualization. The triangular mesh model can be visualized using OpenGL to add 3D manipulation functionality (translation, rotation, scaling) as well as simulating various lighting conditions to view details on the surface of a chromosome. Snapshots of chromosome visualization are provided in Figures 16 and 17, where ambient lighting at different colors and orientations is added to the scene, and a 3D chromosome surface model is viewed at various orientations in different modes (wire-frame/normal).



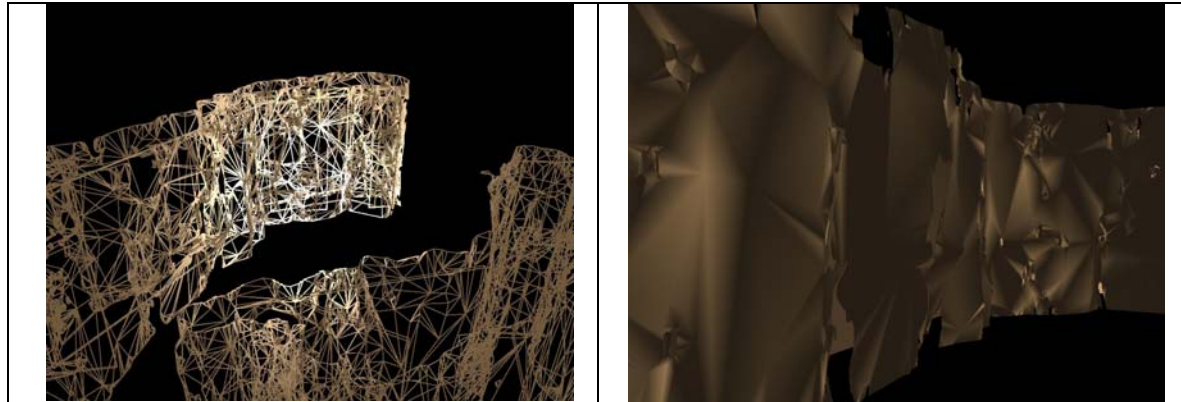
**Figure 13:** View of 3D chromosome model generated by Points2Polys



**Figure 14:** Chromosome splits up towards the top slices



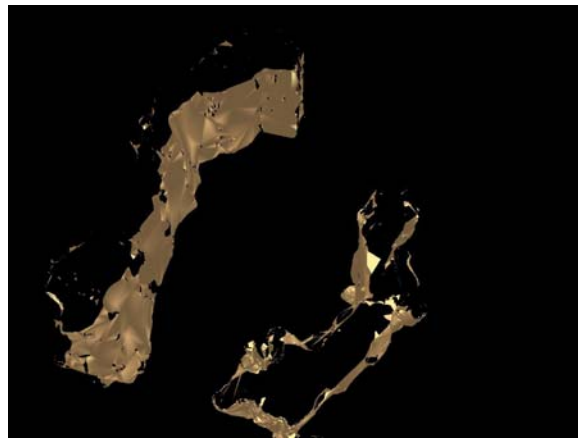
**Figure 15:** (a) A protruding inconsistent surface due to a missed initial guess and refinement of a snake. (b) Original image. (c) Initial position of a snake. (d) Refined contours using snakes.



(a)

(b)

**Figure 16:** (a) Chromosomes in ambient and spot lighting in wire-frame (mesh) mode. (b) The surface of chromosome 1, including surface detail and light reflection.



**Figure 17:** Chromosomes viewed from the top

## 7 Conclusions

In this paper we provide a methodology for an automatic recovery and visualization of a 3D chromosome structure from a sequence of 2D tomographic reconstruction images taken through the nucleus of a cell. Structure is very important for biologists, as it affects chromosome functions, behavior, and the state of the cell. Chromosome analysis is significant in detection of diseases and in monitoring environmental gene mutations. The algorithm incorporates thresholding based on a histogram analysis with a polyline splitting algorithm, shape analysis, and noise removal, contour extraction via active contours, and detection of a 3D chromosome

structure by establishing corresponding regions throughout the slices. Visualization using point cloud meshing generates a 3D surface with a computationally inexpensive and fast approach. The 3D triangular mesh of the chromosomes provides surface detail and allows a user to interactively analyze chromosomes using visualization software.

The ability to capture small features such as constriction points and protrusions that may correspond to the binding of chromosome with adjacent intracellular organelles attests to the accuracy and resolution of our method. The capacity to study the 3D geometry of chromosome structures in an interactive environment is a great asset to physicians and scientists in the diagnosis and treatment of chromosomal abnormalities and the scientific analysis of surface structures of chromosomes such as binding sites with adjacent organelles or intracellular molecules such as proteins or drugs. One of the foremost advantages of our technique is the robustness of visualization based on a fairly small set of input images.

## **8 Acknowledgements**

We wish to thank Prof. John W. Sedat and his laboratory colleagues at the University of California San Francisco for the data, and Dr. William Moss (LLNL) for discussing the problem with us. This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48. UCRL-JC-XXXXXX.

## **9 References**

- [1] B. Lerner, H. Guterman, and I. Dinstein, "A Classification-Driven Partially Occluded Object Segmentation (CPOOS) Method with Application to Chromosome Analysis," in *IEEE Transactions on Signal Processing*, **46**(10), pp. 2841-2847, (1998).

- [2] E. Vidal, and M. J. Castro, "Classification of Banded Chromosomes using Error-Correcting Grammatical Interface (ECGI) and Multilayer Perceptron (MLP)," in *VII National Symposium on Pattern Recognition and Image Analysis*, pp. 31-36, (1997).
- [3] H. Shi, P. Gader, and H. Li, "Parallel Mesh Algorithm for Grid Graph Shortest Paths with Application to Separation of Touching Chromosomes," in *The Journal of Supercomputing*, Kluwer Academic Publishers, Boston, USA, (1996).
- [4] L. Ji, "Intelligent splitting in the Chromosome domain," *Pattern Recognition*, 22:519-532, (1989).
- [5] Y. Gdalyahu, and D. Weinshall, "Flexible Syntactic Matching of Curves and its Application to Automatic Hierarchical Classification of Silhouettes," in *IEEE Conf. on Computer Vision & Pattern Recognition*, 749-764, (June 1998).
- [6] T. McInerney, and D. Terzopoulos, "Deformable Models in Medical Image Analysis: A Survey," in *Medical Image Analysis*, 1(2), (1996).
- [7] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, MIT Press and McGraw-Hill, Boston, USA, pp.194-198, (1995).
- [8] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 673-689, (July 1996).
- [9] M. Kass, A. Witkin, and D. Terzopoulos, "Snake: Active Contour Models," in *Proceedings of First International Conference on Computer Vision*, pp. 259-269, (1987).
- [10] D. J. Williams, and M. Shah, "A Fast Algorithm for Active Contours and Curvature Estimation," in *Image Understanding*, Vol. 55, pp. 14-26, (1991).
- [11] Jeongju, Choi, and J. S. Kim, "Deformable Object Tracking Using Active Contour Model," *15<sup>th</sup> Triennial World Congress*, Barcelona, Spain, (2002).
- [12] L. D. Cohen, "On Active Contour Model and Balloons," in *CVGIP53*. No. 2 pp. 211-218, (1991).
- [13] F. Remondino, "From point cloud to surface: the modeling and visualization problem", *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXIV-5/W10. *International Workshop on Visualization and Animation of Reality-based 3D Models*, Tarasp-Vulpera, Switzerland (Feb 2003).
- [14] R. J. Gardner. *Geometric Tomography*. Cambridge University Press, Cambridge, (1995).

- [15] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara, M. J. Wozny, "Geometrically deformed models: A method for extracting closed geometric models from volume data," in *Computer Graphics (Proc. SIGGRAPH'91 Conf)*, Vol. 25(4), pp. 217-226, (1991).
- [16] L. D. Cohen, and I. Cohen, "Finite element methods for active contour models and balloons for 2D and 3D images," in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15(11), pp. 1131-1147, (1993).
- [17] B. C. Vemuri, and A. Radisavljevic, "Multiresolution stochastic hybrid shape models with fractal priors," in *ACM Trans. on Graphics*, Vol. 13(2), pp. 177-207, (1994).
- [18] T. F. Cootes, A. Hill, C. J. Taylor, D. H. Cooper, and J. Haslam, "The use of active shape models for locating structures in medical images," in *Image and Vision Computing*, Vol. 12(6), pp. 276-285, (July 1994).
- [19] M. M. Dickens, H. Sari-Sarraf, and S. S. Gleason, "A Streamlined Volumetric Landmark Placement Method for Building Three-dimensional Active Shape Models," in *Proceedings of SPIE*, Vol. 4322, pp. 269-280, (2001).
- [20] R. A. Banvard, "The Visible Human Project® Image Data Set From Inception to Completion and Beyond," in *Proceedings of CODATA 2002*, (2002).
- [21] K. R. Subramanian, M. J. Thubrikar, B. Fowler, M. T. Mostafavi, and M. W. Funk, "Accurate 3D reconstruction of complex blood vessel geometries from intravascular ultrasound images: *in vitro* study," in *Journal of Medical Engineering and Technology*, Vol. 24(4), pp. 131-140, (2000).
- [22] B. Lerner, M. Levinstein, B. Rosenberg, H. Guterman, L. Dinstein, and Y. Romem, "Feature selection and chromosome classification using a multilayer perceptron neural network," in *IEEE International Conference on Neural Networks*, Vol. 6, pp. 3540-3545, (1994).
- [23] J. M. Keller, P. Gader, O. Sjahputera, C. W. Caldwell, and H.-M. T. Huang, "A fuzzy logic rule-based system for chromosome recognition," in *Proceedings of the Eighth IEEE Symposium on Computer-Based Medical Systems*, pp. 135-132, (1995).
- [24] M. R. Arnison, C. J. Cogswell, N. I. Smith, P. W. Fekete, and K. G. Larkin, "Using Hilbert transforms for 3D visualization of differential interference contrast microscope images," in *Journal of Microscopy*, Vol. 199, pp. 79-84, (2000).
- [25] P. Engelhardt, J. Ruokolainen, A. Dulenc, L. G. Överstedt, H. Mehlin, and U. Skoglund, "3D-reconstruction by electron tomography (EMT) of whole-mounted DNA-depleted metaphase chromosomes show scaffolding macro coils, 30-nm fibers and 30-nm particles," in *International Conference on 3D Image Processing in Microscopy*, (1994).