Towards Recovery of 3D Chromosome Structure

Sabarish Babu, Pao-Chuan Liao, Min C. Shin
Dept of Computer Science
UNC Charlotte
Charlotte, NC 28223
{sbabu, pliao, mcshin}@uncc.edu

Leonid V. Tsap
Electronics Engineering Dept
Lawrence Livermore National Laboratory
Livermore, CA 94551
tsap@llnl.gov

Abstract

[1] *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.*

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 Kleinfelter's, Down's, and Turner's syndrome. 3D visualization of the chromosome can be useful for biologists in the following ways: (1) identifyng 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]. Lerner et al. [9] 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. 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 [11].

3D visualization enables scientists to discern occluding chromosomes for further classification better than 2D image analysis. There have also been works on visualization and 3D reconstruction of large and small biological objects based on various imaging

[1] 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-PROC-203893

modalities. For example, Engelhardt et al. visualized metaphase chromosome from human (HeLa) cell lines using electron microscopy (EM) [10]. 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

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 four 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 [4].
2. Noise removal is achieved by connected component labeling (CCL) [4] 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 [6].
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. [5] to correspond regions of the same chromosome between slices. This method achieves tracking even when the chromosome breaks into multiple regions in some slices.

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 [4]. This set is taken for each chromosome, to create point clouds. Then, using meshing technique [8], we construct a mesh representing the surface of each chromosome.

3. Region Segmentation and Noise Removal

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 [4]. It iteratively divides a curve into a set of line segments denoted by a set of vertices (see Figure 1). 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 1a.) When a local minima is detected, we find the threshold by finding the local minima (k) with the highest peakiness [4]. The peakiness is defined as

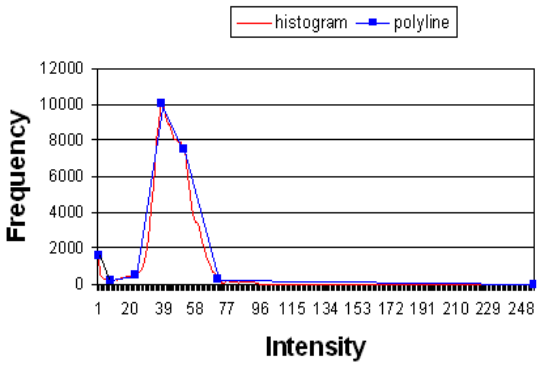
$$\min\{H(i), H(j)\} / H(k) \qquad (3.1)$$

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.

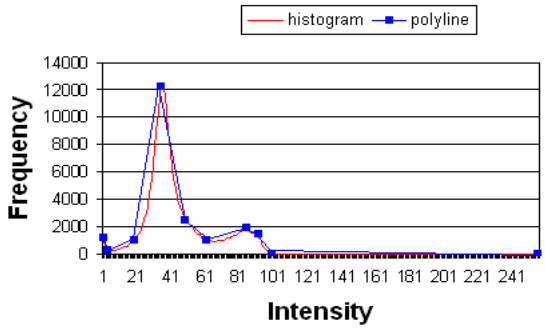
In Figure 1, no local minima is present on the histogram for slice 9 (Figure 1a), while one local maxima is found at intensity 37, thus indicating that the histogram for slice 9 is unimodal. After applying polyline splitting to the histogram contour of the entire sequence, slices with unimodal histogram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

Figure 1b 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 2 represents the results of polyline splitting of the histogram contour, applied to the original image (Figure 2a) 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). Table 1 shows results for every tenth slice followed by threshold values and peakiness estimates.

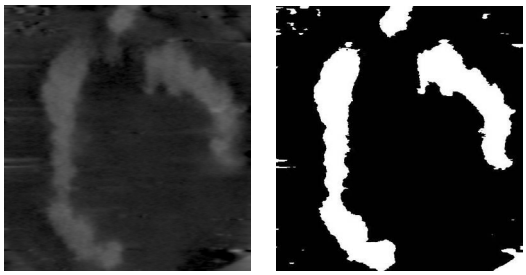


(a) Slice 9 with unimodal histogram



(b) Slice 30 with bimodal histogram

Figure 1: Histograms from images with different modality.



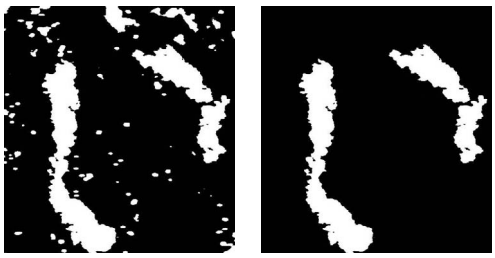
(a) (b)

Figure 2: (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.

After region segmentation, small region removal is performed to eliminate noise in the thresholded image. Regions are detected by using connected component labeling [4] and the regions smaller than 500 pixels are removed (refer to Figure 3).

slice	peakiness	threshold
1	0.00	not found
10	1.61	73
20	0.00	not found
30	1.76	58
40	2.01	58
50	1.53	64
60	0.00	not found

Table 1: The computed threshold for every tenth slice is shown. Note that a threshold is not found for slices with the maximum peakiness of 0, indicating that those histograms are unimodal.



(a) (b)

Figure 3: (a) Slice 53 after region segmentation and (b) after small region removal.

4. 2D Contour Refinement

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

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

where v(s) = (x(s), y(s)), E_{int} indicates the internal force of the spline due to bending, E_{image} denotes the image influence, and E_{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 [7] due to speed and stability considerations.

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.

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 4.

If the initial threshold estimate is incorrect, then the algorithm does not perform optimally, and improper node placement can result. Figure 5 is extracted from the same slice as the one in Figure 4, but with a lower T_{low} at 60. The detected edges with lower T_{low} have moved slightly away from the center when compared to Figure 4. 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 5, 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.

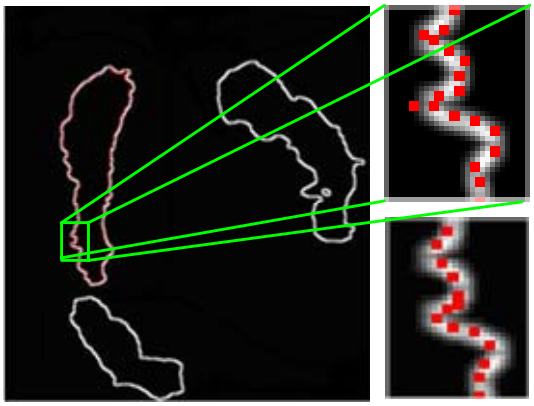


Figure 4: (Slice 23, T_{low} = 61, T_{high} = 110) Initial snake position (top-right), and snake in a stable position with minimal energy (bottom-right).

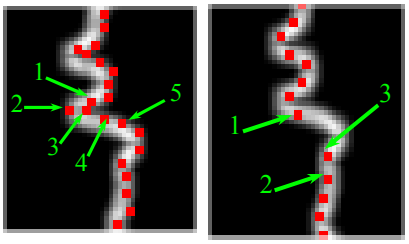


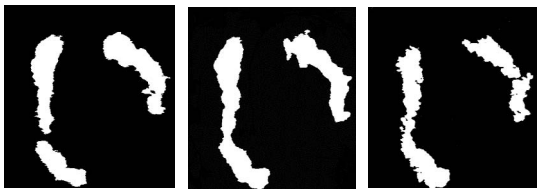
Figure 5: (left) An initial position of a snake in an image with an incorrect initial threshold selection (the same slice as in Figure 4). (right) A snake in a stable position with a gap.

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. [5] 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 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 6 represents two instances of segmented images from end slices (the left image is from slice 23, and the right image 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.



(a) Slice #23 (b) Slice #38 (c) Slice #53

Figure 6: 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 [5] 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:

1. An instance of **correct** classification is specified when a pair of regions in the adjacent images has at least $\boldsymbol{T}$ percent of the pixels in the chromosome region R_k in the first image marked as pixels in chromosome region R_l in second image. This follows from the premise that $R_k \bigcap R_l \neq 0$ for all k and l, if $k = l$.
2. An instance of **over-segmentation** classification is specified when a chromosome region R_k in the first image and a set of regions in the second image R_l^1 to R_l^n, has at least $\boldsymbol{T}$ percent of the pixels in each chromosome region R_k in the first image marked as pixels in the union of chromosome regions R_l^n of the second image.
3. An instance of **under-segmentation** classification is specified when a set of chromosome regions in the second image R_l^1 to R_l^n and a chromosome region R_k in the first image has at least $\boldsymbol{T}$ percent of the pixels in chromosome region R_k in the first image marked as pixels in the union of chromosome regions R_l^1 to R_l^n in the second image.

Applying the metric described here 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. 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.

6. 3D Visualization

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.

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 7. It shows the point at which the chromosomes split up toward the end slices (the centromere), which can be observed in the 3D model. Figure 8 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 7 and 8, 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.

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, centromere, 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.

Acknowledgements

We wish to thank Professor 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.

8. 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 Trans 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] R. Jain, R. Kasturi, and B. G. Schunck, *Machine Vision*, MIT Press and McGraw-Hill, Boston, USA, pp.194-198, (1995).

[5] 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 Trans on PAMI,* pp. 673-689, (July 1996).

[6] M. Kass, A. Witkin, and D.Terzopoulos, "Snake: Actve Contour Models," in *Proceedings of First International Conference on Computer Vision*, pp. 259-269, (1987).

[7] D. J. Williams, and M. Shah, "A Fast Algorithm for Active Contours and Curvature Estimation," in *Image Understanding*, Vol. 55, pp. 14-26, (1991).

[8] 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).

[9] 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).

[10] 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).

[11] 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).

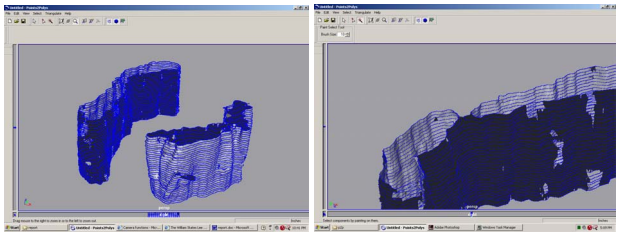
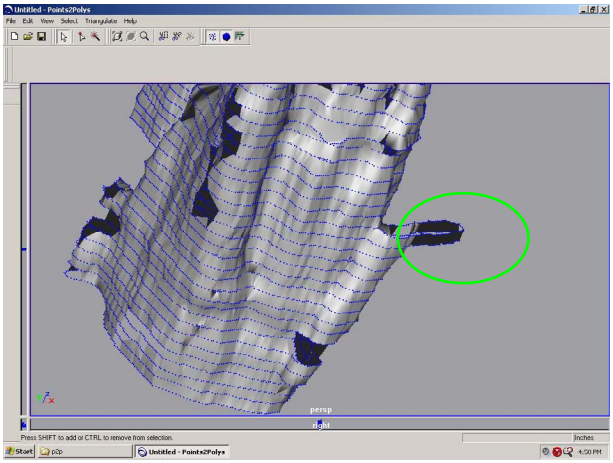


Figure 7: View of 3D chromosome model generated by Points2Polys. Note that the Chromosome splits up towards the top slices



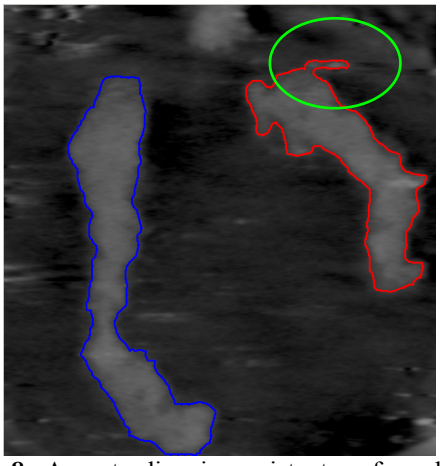


Figure 8: A protruding inconsistent surface due to a missed initial guess and refinement of a snake (top). Refined contours using snakes (bottom).