Spring 2009

Development of a diaphragm tracking algorithm for megavoltage cone beam CT projection data

Mingqing Chen
University of Iowa

Recommended Citation
Chen, Mingqing. "Development of a diaphragm tracking algorithm for megavoltage cone beam CT projection data." MS (Master of Science) thesis, University of Iowa, 2009.
https://doi.org/10.17077/etd.ib1yibr4

Follow this and additional works at: https://ir.uiowa.edu/etd
Part of the Electrical and Computer Engineering Commons

DEVELOPMENT OF A DIAPHRAGM TRACKING ALGORITHM FOR MEGAVOLTAGE

CONE BEAM CT PROJECTION DATA

by

Mingqing Chen

A thesis submitted in partial fulfillment
of the requirements for the Master of Science degree in
Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2009

Thesis Supervisor: Assistant Professor R. Alfredo C. Siochi

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Mingqing Chen

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Electrical and Computer Engineering at the May 2009 graduation.

Thesis Committee: R. Alfredo C. Siochi, Thesis Supervisor

Milan Sonka

Xiaodong Wu

Gary Christensen

To My Parents, Jian Chen and Min Lu
To Jenny

ACKNOWLEDGMENTS

First of all, I would like to express my sincere gratitude to my adviser Dr. Siochi. He has established the software framework and had the overall idea about the project before I came here. He guided me into the field of image guided radiation therapy and this challenging project which aimed at clinical application by his wealthy knowledge and brilliant ideas. And I greatly appreciate his constructive advice and patient tutoring during the research. Without his guide I could not achieve this goal.

I would like to thank Professor Sonka, who gives me the enlightening idea on detecting diaphragm in 200 frames simultaneously. I am grateful to Professor Wu and Professor Garvin for their help on 3D image segmentation. Without discussion with them, the work would not have been succeeded like this. My special thanks go to Qi Song for providing the code library of optimal graph search and his explanation on the previous work of graph search area. I would also like to thank Kai Ding, Xin Dou, Kunlin Cao, Ying Wei and Yunlong Liu for their friendship and help on many technical problems of this work.

The contributions of all these people are greatly appreciated.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

Background and Motivation

In radiotherapy, lung tumor localization is of great concern due to unavoidable respiratory motion. Respiratory gating techniques have been developed over the years to increase the radiation dose to the tumor while reducing radiation to the adjacent healthy tissues. In our clinic, an abdominal belt is used as an external marker to record the correlation between the phase and location of the tumor during the kilo-voltage (KV) treatment planning CT scan. During each treatment, the cranio-caudal position of the tumor will be determined by the pressure on a strain gauge placed under an abdominal belt. Physicians can determine when to turn the beam on or off based on tumor position.

Although various attempts have been made to quantify the correlation between external abdominal signals and tumor motion [35, 1], it is reported that various respiratory phase delays occur for external markers for patients having impaired lung function [20, 26], which makes the external marker not fully reliable. These studies are based on the assumption of direct tumor-diaphragm relationship. If we can accurately locate the position of the diaphragm apex, it will be a more powerful method of characterizing respiration, since the amplitude of motion can be determined, while external markers such as strain gauges only provide relative phase information [32].

With the recent development of megavoltage cone beam CT (MVCBCT) systems, accurate daily localization can be derived by registering the reconstructed 3D MV CT with the planning KV CT before each treatment. The changes in tumor shape, size and position during the course of treatment can be monitored by registering daily localization 3D MVCBCT images with planning KV CT images [25]. Studies on the feasibility of tracking objects in MV portal images suggested that it is possible to track diaphragm motion in MVCBCT raw projection images [21]. In our current clinic, physicians should identify the position of the diaphragm manually for each projection image. This is time consuming which made it impossible to apply clinically as there is a strict time limit for patient localization before treatment. Xu et al. developed an optical flow method to track lung tumors in fluoroscopic videos [38]. However, a tumor contour in the initial reference frame is needed to be drawn by a clinician or transferred from digitally reconstructed radiographs. Keatley et al. developed a diaphragm tracking method using active contours with dynamic programming on fluoroscopic sequences taken from one gantry angle [17]. But the diaphragm contour is only a local energy minima and the process also needs a contour and several landmarks drawn by humans. In this work, a computer automated diaphragm tracking algorithm is developed and tested on fluoroscopic sequences taken with the camera rotating about the patient. The algorithm is guaranteed to find the global optimum with fewer human interactions. The result is proved to be robust and accurate. The running time satisfies the clinical application.

The sequence of projection images has the same dynamic features as a video. We can fully utilize the algorithms of video processing to solve our problem. The following part of this thesis will give a brief review on several important branches of video tracking methods. In chapter 2, a brief introduction of MVCBCT projection image and our software platform will be introduced. Then I will explain the previous work by Siochi on calculating the region of interest (ROI) of the diaphragm apex using CBCT geometry

[29]. Then techniques experimented in this work will be described, including image filters, dynamic programming (DP) with shortest path, principle component analysis, active shape model (ASM), and optimal graph search. Dyanmic Hough transform, the method achieving the best result in this work, will be described with emphasis. In chapter 3, results of the different methods will be illustrated and discussed, including DP with shortest path in 2D, ASM, optimal graph search, optimal graph search with edge cost and dynamic Hough transform. Future work and unresolved problems are also included in that chapter.

An Overview of tracking methods in an image sequence

A video is a sequence of images. Each image within this sequence is called a frame. Video tracking, sometimes referred to as motion tracking, motion estimates or visual tracking in other literature, is an automatic algorithm to locate single or multiple moving objects in an image sequence. Automation in various areas such as video surveillance and servo, facial recognition, medical imaging and automatic inspection requires fast and robust video tracking algorithms. For example, Frank et al. developed a traffic video analysis system that can track vehicles on downtown roads with recognition of occlusion situations [11]. Harrell et al. developed a fruit-tracking method to estimate the size and position of a valid fruit region in real time images to control the motion of a fruit-picking robot [12]. Boyer applied video tracking techniques to reconstruct 3D object models from image sequences [6].

Ideally, the image sequence of a video is continuous. For a digital device, small time intervals between adjacent frames exist which makes objects move a small distance from one frame to the next. This property is utilized by all the video tracking algorithms. It is either represented by smoothness constraints or the speed of motion. The main difficulty for video tracking is that there is variation in object location and object deformations. The illumination may also change over the frames, which leads to changing gray values of the pixels of the same region in two frames. Partial or full occlusion of the object may also exist which will cause deviations to object tracking. The video tracking algorithm can be generally categorized into two classes. The first is to locate and track objects frame by frame throughout image sequence. The second is to track objects in all frames simultaneously. Most video tracking algorithms belong to the first category. They aim to find a correlation between the adjacent frames. Optical flow describes this correlation by displacement velocity. It approximates image motion based on the hypothesis that the intensity structure of the same image regions are constant for a short duration [5], and it assumes that every small block within the reference frame has a velocity to the next frame. A least-square technique is used to calculate this velocity field, by minimizing a squared error under restriction of first-order derivatives of the optical flow constraint equation [21], or by solving equations of velocity using second-order derivatives [24], or by solving velocity equations with combination of local estimates [28]. A detailed introduction and comparison of different optical flow techniques can be found in [4].

Similar to optical flow, parametric models also compute the spatial variation of two adjacent images. However, it assumes that the optical flow within a region can be represented by a low-order polynomial. This assumption is very close to reality when the motion between frames is mainly caused by camera motion, without displacement of the local individual object. The parametric model is chosen based on the estimate of global motion, such as the six-parameter affine model, eight-parameter quadratic model, eight-parameter perspective model, etc. The parameters are calculated based on the minimization of an error metric, such as squared differences between reference frame and transformed moving frame. An initial motion estimation is needed and an iterative scheme using gradient descent or other optimization technique is used to update the parameters to achieve a local minimum [19, 22].

Instead of computing displacement for a region, objects in an image sequence can be represented by object features, such as an object boundary. Thus, object tracking can be achieved by contour tracking. Snake, i.e. parametric active contour model [16], is an edge detection technique that locates a spline in large gradient pixels in an image with an iterative energy minimization scheme. Given an initial shape, the spline will be adjusted at each iteration in nearby regions using Euler’s method. The energy includes external constraint forces to attract the spline to the object boundary and internal forces to control the smoothness of the curve. As traditional snake methods can only find a local minimum of splines near the initial position, many derivative methods of active contour have been developed. Xu modified the external force of the snake energy equation by computing a gradient vector flow field, enabling a larger capture range. The spline can converge to the expected position even if the initial shape is far away [37]. Davi et al. developed a dynamic programming scheme to minimize the energy of the contour with a global minimum guaranteed. And a multi-scale strategy is used to speed up the search time [10].

In applications in video tracking, a contour will be detected first in the reference frame and the result will be the initial contour for the moving frame.

Sometimes the shape pattern of objects to be tracked is known. We can utilize this knowledge as prior information before video tracking. A large set of training data in sampled images needs to be derived to get this statistical information. ASM belongs to this family and is also a contour tracking algorithm. It uses sets of labeled points to represent shapes of objects. The shape in a 2D image represents a point in a higher dimensional shape vector space. It detects contours of objects with prior shape restrictions computed from training data [34]. The contour to be detected is updated to the position closer to the edge boundary using an iterative searching scheme. It will be confined into an area near the mean of training contours. Principle Component Analysis (PCA) is used to save memory and time while retaining most of the training information [34]. Both the advantage and disadvantage of ASM is discussed in [18] with “more active shape model” (MASM), an improved technique of ASM.

The other category of video tracking algorithms is to track objects in all the frames simultaneously. Kang et al. developed an optimal graph search algorithm for 3D image segmentation. Instead of iterative scheme searching for a local extremum, optimal graph search computes a surface within a 3D graph directly with a global minimum of the sum of costs of nodes [15]. The surface is guaranteed to be continuous under the restriction of the smoothness constraint. The problem of computing this surface is converted to finding the maximum-flow between two nodes. There are various algorithms that calculate the maximum-flow. A pseudo-flow algorithm developed by Hochbaum [14] and the algorithm developed by Boykov and Kolmogorov [39] both have the advantage of saving time and memory and are proved to be superior to traditional maximum-flow algorithms such as the Ford-Fulkerson [9] and push-relabel algorithms [2].

To sum up, the design of a video tracking method requires consideration in all aspects. For example, is an iterative scheme used or not? Is the local optimum or global optimum guaranteed? What kind of image filter is used? Is prior information needed? All these choices depend on the property of the video image sequence to be processed. For example, is the major component of the motion in the image global or local? Is there a general shape pattern for the objects to be tracked? Can the objects to be tracked be represented by contours? What is the image quality, i.e. what is signal to noise ratio (SNR)? In this work, different techniques are tested including dynamic programming with path detection, ASM, optimal graph search and parametric shape model with dynamic programming. The goal is to develop an algorithm that best fits our special application of diaphragm tracking in MVCBCT projection images. The next section will analyze the characteristics of these fluoroscopic images.

CHAPTER TWO

MATERIAL AND METHODS

Hardware and Software Platform

In our case, the global motion of the projection image sequence of MVCBCT is caused by rotation of the gantry and the image detector. Using the standard cone beam protocol, the gantry rotates over 200 degrees from 270° to 110°. Figure 1 is a picture of the MVCBCT device used in our clinic. Instead of traditional parallel or fan beams, the linac X-ray source produces a cone of rays with its base on the detector and its apex on the source. The set of open field 2D projection images are acquired at different positions around the patient. The data are acquired with a 2D detector array. The imaging process does not need table movement.

The 2D detector is an amorphous Si flat panel electronic portal imaging device (EPID) with resolution of 0.4 × 0.4 mm^2. It has an imaging area of 27.4 × 27.4 cm^2 at the patient plane and an image size of 1024 × 1024, providing a pixel size of 0.027 cm. This resolution is enough for the motion being tracked. The distance between the source and isocenter (Source to Axis Distance, SAD) is 100 cm and 145 cm for source and EPID (SID) [31]. The EPID acquires one 2D projection image per degree of rotation. In the clinic, 200 raw projection images are acquired over 200 degrees while the gantry rotates in order to have sufficient data for 3D cone-beam reconstruction. So each frame is one image projected from a certain angle.

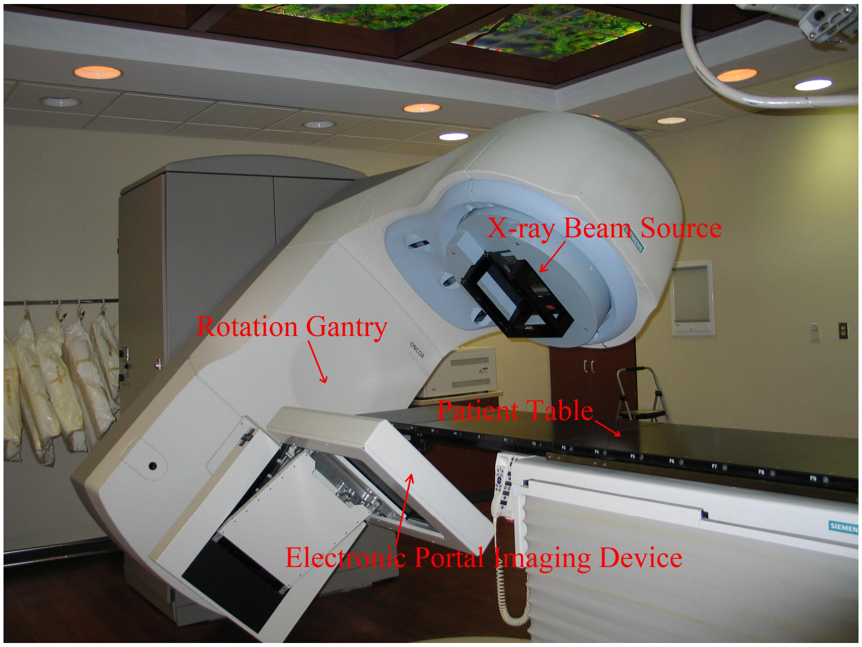

Figure 1: Siemens Mvision MVCBCT device used in the Department of Radiation Oncology in the University of Iowa Hospital

The local motion is the respiratory displacement of the lung and other nearby organs such as the liver and the heart. The diaphragm has a general curve pattern opening up in the inferior direction and it can be delineated by a contour. It will be discussed in later paragraphs that this general curve pattern of the diaphragm can be well represented by a parabola. The contrast of MV CBCT images is relatively low compared to KVCT because Compton scattering provides the majority of the beam attenuation [25]. And there are many confounding objects near the diaphragm like the contralateral lung, heart, spine, and treatment table which will cause deviation to the tracking process.

The total imaging time is 50 seconds which is 0.25s for each view. Typically a patient with training takes 16 complete cycles to be observed with 200 views [31]. The range of diaphragm motion in the superior-inferior (SI) direction is between 0.5 and 3.0

cm. Using the respiratory frequency, the speed of diaphragm motion can be calculated by the following equation:

$$v_{frame} = \frac{R}{N_{1/2}} \cdot \frac{SAD}{SID} \tag{2.1}$$

Where R is the displacement of the diaphragm apex between full exhale and full inhale frames. $N_{1/2}$ is the number of frames for a half respiratory cycle. With this information, we can get a rough estimate of the diaphragm motion to be tracked. For example, if there is a typical patient with 16 cycles during the imaging process and $R = 0.75cm$, 12.5 frames for each cycle, i.e. $N_{1/2} = 6.25\,frames$ can be derived. So $v_{frame} = 0.12cm/\ frame$, i.e. $4.44\,pixel/\ frame$. Note that this calculation is based on two approximations. One is that the diaphragm motion is assumed to be at SAD distance to the X-ray source. The second is that this equation leads to the average speed. But in fact, the diaphragm cannot be exactly at SAD distance and the motion is slow at full inhale and full exhale positions while fast in the intervals between these positions.

Dr. Siochi has developed a software application called Just Enough Dicom (JED) before this work. The software can read DICOM files of projection images imported from MVCBCT. It is developed in Microsoft Visual Basic 2008 and it provides a Windows interface that allows users to do multiple operations such as reading, viewing the images and analyzing the diaphragm tracking result. All of the diaphragm tracking techniques in this work is implemented on this platform. A typical interface of JED is shown in Figure 2. The projection image is shown on the right. On the left there are grids representing coordinates. And two diagrams of position of diaphragm apex are shown. The top one is in the SI direction while the bottom one is in the horizontal direction in the image plane. There are menu bars on the top which controls the view and diaphragm tracking. The status bars are at the bottom indicating the frame and pixel information.

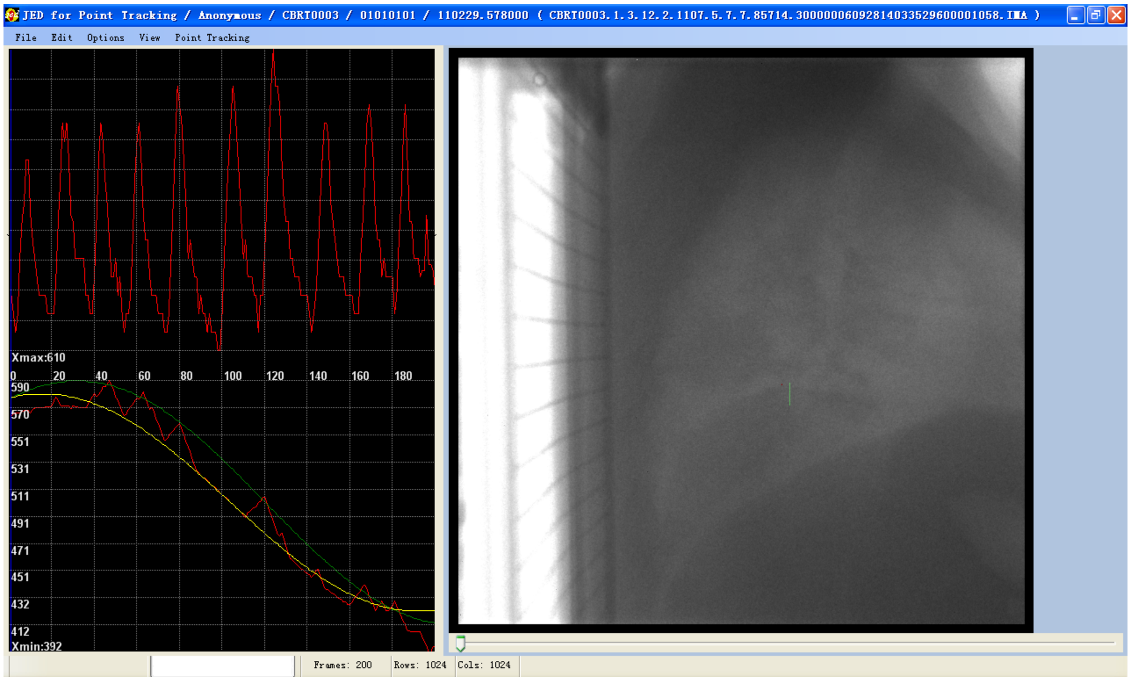
JED for Point Tracking / Anonymous / CBRT0003 / 01010101 / 110229.578000 (CBRT0003.1.3.12.2.1107.5.7.7.85714.30000006092814033529600001058.IMA)
File
Edit
Options
View
Point Tracking
Xmax:610
0
20
40
60
80
100
120
140
160
180
590
570
551
531
511
491
471
451
432
412
Xmin:392
Frames: 200
Rows: 1024
Cols: 1024

Figure 2: Windows interface of Just Enough Dicom (JED)

Computation of ROI

A projection image covers a large area, including at least one of the lungs and its adjacent organs. It is unnecessary and time consuming to search through the whole image for the diaphragm apex. A robust method to locate the ROI is needed. In this section the technique of computing the ROI of the diaphragm apex developed by Siochi is described [32, 29]. In Siochi's method, four positions of the diaphragm apex are needed to calculate the ROI: two full inhale and two full exhale in the corresponding respiratory cycles. The two cycles are recommended to be about 90° apart. Positions of the diaphragm apex in full inhale and full exhale in 3D that define the range of diaphragm apex motion are computed from these four points based on CBCT geometry. Then these two points are projected into every 2D frame and a rectangle defined by the two points for each frame is derived. This rectangle is the ROI for the diaphragm apex for each frame. The procedure of this method is illustrated in the following figure:

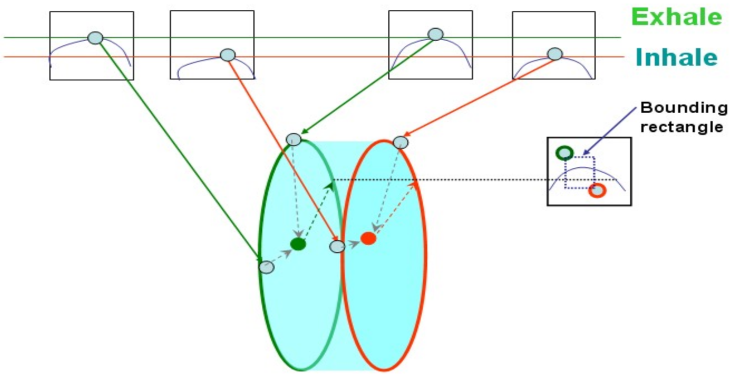

Figure 3: The illustration of computing ROI for each frame

Image from [30]

The CBCT geometry used in this method is illustrated in Figure 4. It is a picture showing that both the X-ray source and image detector rotate about the isocenter, which is considered as the origin of the 3D coordinates. The X and Y axes are in the paper plane and the Z axis is perpendicular to the paper plane. First the user identifies the projected diaphragm apex in the flat image detector with 3D coordinates (x_o, y_o, z_o). The horizontal distance from this point to the center of the image detector is R_o. The center of the image detector is actually the projected isocenter in this frame. The first step is to project user identified points onto a plane called the minified view at SAD. It is a plane parallel to the detector and has the distance of SAD to the source. The projected point is (x_p, y_p, z_p) which is the intersection of this plane and the line through the source and (x_o, y_o, z_o). The distance from (x_p, y_p, z_p) to isocenter is R_p. From simple trigonometry, (x_p, y_p, z_p) and R_p can be derived from (x_o, y_o, z_o) and R_o using following equations:

$$\begin{aligned}
&R_{p1} = R_{o1}\frac{SAD}{SID}; R_{p2} = R_{o2}\frac{SAD}{SID} \\
&x_{p1} = R_{p1}\cos\theta_1; y_{p1} = R_{p1}\sin\theta_1; x_{p2} = R_{p2}\cos\theta_2; y_{p2} = R_{p2}\sin\theta_2 \\
&z_{p1} = z_{o1}\frac{SAD}{SID}; z_{p2} = z_{o2}\frac{SAD}{SID}
\end{aligned} \tag{2.2}$$

Where θ_1 and θ_2 are rotation angles between the image plane and the X axis illustrated in Figure 4. The subscripts 1 and 2 correspond to two frames in different respiratory cycles (For conciseness, Figure 4 only shows coordinates in one frame). Now two points at the minified view at SAD are calculated and the position of the source can be calculated as:

$$x_{s1} = -SAD\cdot\sin\theta_1; y_{s1} = SAD\cdot\cos\theta_1; x_{s2} = -SAD\cdot\sin\theta_2; y_{s2} = SAD\cdot\cos\theta_2; \tag{2.3}$$

Now for each frame, a line can be determined by the source and point in minified view. Ideally, if the position of diaphragm apex will reach the same position in every full exhale and inhale time, the lines in these two frames will intersect with each other. However, there is some irregularity and movement during respiration even for a trained patient. And the motion is sampled at 0.25s by MVCBCT. These factors lead to the fact that the two lines are not coplanar. In order to calculate the position of the diaphragm apex in 3D, we project these two lines in the paper plane and calculate the point of intersection in 2D to determine x_d and y_d. For the Z direction, we use the average of the two values of z_d at (x_d, y_d) along lines. So (x_d, y_d, z_d) can be calculated as:

$$
\begin{aligned}
& m_1 = \frac{x_{p1} - x_{s1}}{y_{p1} - y_{s1}}; m_2 = \frac{x_{p2} - x_{s2}}{y_{p2} - y_{s2}} \\
& x_d = x_{s1} + m_1(y_d - y_{s1}); y_d = \frac{x_{s2} - x_{s1} - m_2 y_{s2} + m_1 y_{s1}}{m_1 - m_2}; \\
& y_{f1} = -x_d \sin\theta_1 + y_d \cos\theta_1; y_{f2} = -x_d \sin\theta_2 + y_d \cos\theta_2 \\
& z_{d1} = \frac{(SAD - y_{f1})}{SAD} z_{p1}; z_{d2} = \frac{(SAD - y_{f2})}{SAD} z_{p2}; z_d = \frac{z_{d1} + z_{d2}}{2}
\end{aligned}
\tag{2.4}
$$

Where m_1, m_2, y_{f1}, y_{f2} are temporary variables and m_1 and m_2 are the inverse of the slopes of the two lines. y_{f1} and y_{f2} are the distances of (x_d, y_d) to the plane on the minified view. So the actual positions of diaphragm apex during full inhale and full exhale are derived. For each frame, if we project these two points to the detector plane, we can use the projected points as the diagonal vertices of a rectangle. This rectangle is exactly the ROI we want. For an arbitrary frame with rotation angle θ, the coordinates of the projected points can be calculated as

$$
\begin{aligned}
& x_s = -SAD \cdot \sin\theta; y_s = SAD \cdot \cos\theta \\
& m_s = \frac{x_s - x_d}{y_s - y_d} \\
& y_f = -x_d \sin\theta + y_d \cos\theta \\
& R_p = \frac{x_d - m_s y_d}{\cos\theta - m_s \sin\theta}; \\
& R_o = R_p \frac{SID}{SAD}; z_o = z_o \frac{SID}{SAD - y_f}
\end{aligned}
\tag{2.5}
$$

Where (R_o, z_o) is the projected position of diaphragm apex in full exhale or full inhale in the image plane. Note that as the origin for 3D coordinates is the isocenter, the origin in the detector plane is the center of the image, i.e. the isocenter projected on the detector plane.

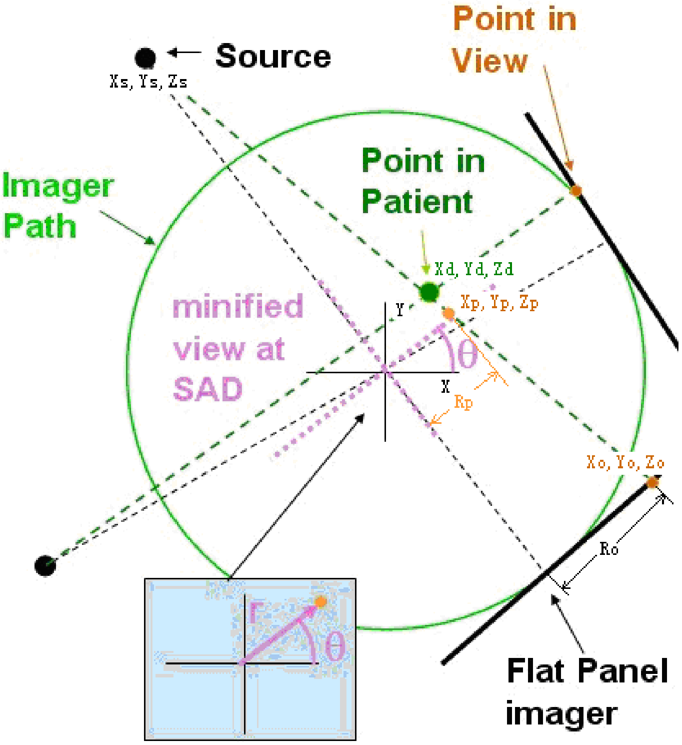

Figure 4: CBCT geometry for calculating ROI

Image modified from [30]

Image Filtering

The following figure shows one frame of the original MVCBCT projection image in the region around the ROI in one frame.

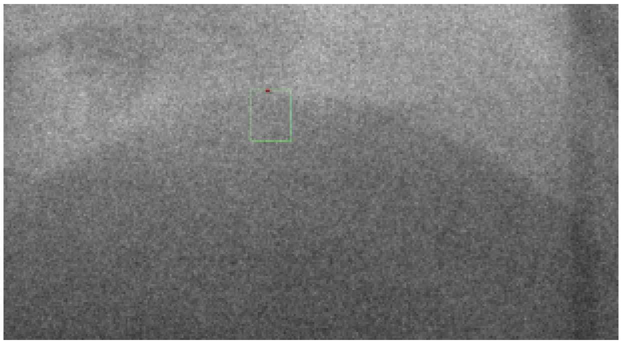

Figure 5: Region near ROI (green rectangle) in a projection frame displayed in JED

It can be observed that the CNR of MVCBCT is relatively low compared to that of traditional CT images. An approximate contour of the diaphragm can be seen by the eye, but the edge is blurred. There is no distinct contrast between bright lung area and the dark area below. And the noise and other interfering objects in the image makes diaphragm tracking more difficult. Image preprocessing techniques are required to enhance the diaphragm contour while reducing noise and other interference. The selection of image filters depends on the requirement of applications. The balance between the quality of filtering and the running time is always a matter of concern. Convolution kernels such as average filters or Gaussian filters [38] are widely applied in image processing applications with low computation and good noise repression characteristics. However the structure of interest such as an edge is also blurred. Statistical filters such as Median smoothing preserves edges [17], but the sorting process takes much time. Nonlinear filters such as anisotropic diffusion and curvature flow are also edge-preserving. However, these filters are more computationally expensive [27]. Generally, the more features of interest a filter can preserve, the more time it will consume. In this section, Gaussian and anisotropic diffusion filters implemented in JED will be described and compared. Then the technique for enhancing the diaphragm contour used in JED will be discussed.

A Gaussian filter has a Gaussian function as its filter window. For 2D images, it has a kernel with the following mathematical expression:

$$g(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2}{2\sigma^2}}\cdot e^{-\frac{y^2}{2\sigma^2}} \tag{2.6}$$

Equation 2.6 also shows that a 2D Gaussian filter can be decomposed into the product of two 1D Gaussian filters. This property enables us to decompose the 2D convolution into two 1D convolutions, which will save much computation time. For example, assume that the size of the Gaussian filter is $N\times N$. For each pixel, the 2D convolution involves $N\times N$ multiplications and $N\times N-1$ additions. With the decomposition, each 1D convolution only requires N multiplications and $N-1$ additions. The sum is $2N$ multiplications and $2N-2$ additions.

The convolution of Gaussian filters can be replaced by multiplication in the frequency domain using Fast Fourier Transform (FFT) and its inverse. However, FFT will be significantly faster than the convolution only when N is sufficiently large.

Anisotropic diffusion filters encourage intra-region smoothing while inhibiting inter-region smoothing. It uses an evaluation function with locally adaptive diffusion strengths to decide whether an edge should be smoothed or enhanced. It has the mathematical expression:

$$\frac{\partial}{\partial t}I(x,t)=\nabla(c(x,t)\nabla I(x,t)) \tag{2.7}$$

Where $I(x,t)$ is the intensity of a MVCBCT projection image, x refers to the image coordinate (R,z) , t refers to the iteration step and $c(x,t)$ is the diffusion evaluation function which monotonically decreases with the image gradient magnitude.

From this equation we can observe that the anisotropic diffusion uses an iterative scheme that calculates an update to the pixel intensity each time. The evaluation equation is:

$$c(x,t) = f(|\nabla I(x,t)|) \qquad (2.8)$$

It allows for locally adaptive diffusion strengths. Many monotonically decreasing continuous functions suffice. The one applied in JED is:

$$c(x,t) = \exp(-(\frac{|\nabla I(x,t)|}{K})^2) \qquad (2.9)$$

Where K is a diffusion constant and it decides the behavior of the filter. It should be chosen carefully to correspond to gradient magnitudes produced by noise and less than the gradient magnitude of the edges. The function is plotted as follows:

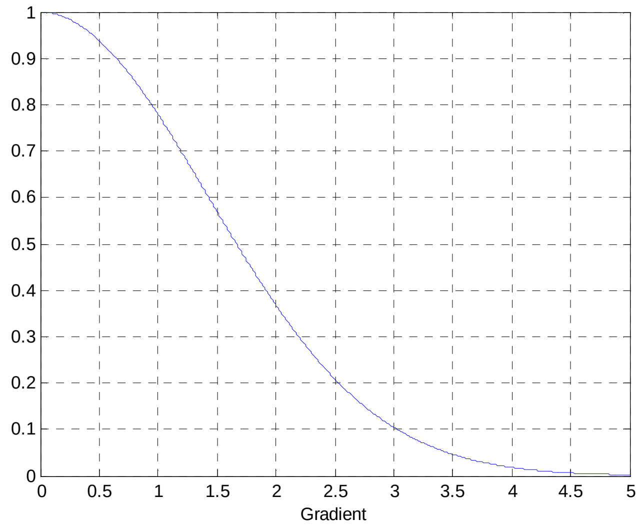

Figure 6: Plot of diffusion evaluation function of anisotropic diffusion applied in JED

If we combine both $c(x,t)$ and $\nabla I(x,t)$, we can define a flow function as:

$$\Phi(x,t) = c(x,t)\nabla I(x,t) \tag{2.10}$$

Then we can rewrite the equation of anisotropic diffusion as:

$$\frac{\partial}{\partial t} I(x,t) = \nabla(\Phi(x,t)) \tag{2.11}$$

The flow function is plotted as a function of the gradient as follows:

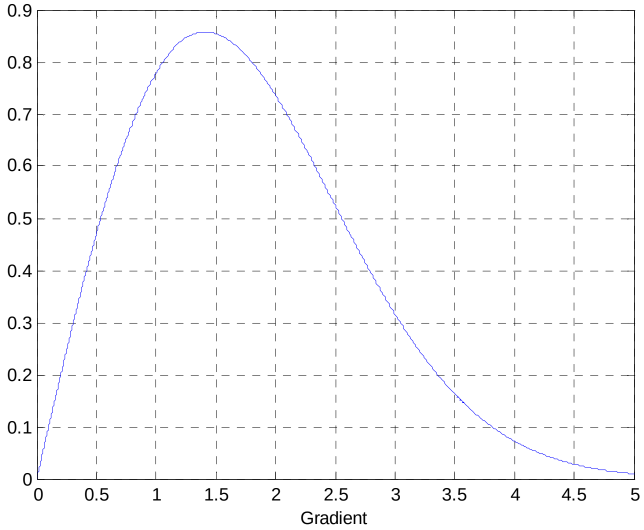

Figure 7: The flow function of the anisotropic diffusion filter applied in JED

It can be observed that the flow increases with the gradient strength to the point where $\nabla I \approx K$, then decreases to zero. This behavior implies that the diffusion process will increase the intensity of pixels with $\nabla I < K$ as the derivative of the flow function is positive. The intensity will be reduced for the pixels with $\nabla I > K$ as the derivative of the flow function is negative. Noise will be removed gradually as the intensity of pixels is updated iteratively. Typically 5 to 20 iterations are needed to evolve the raw projection image into a smoothed and enhanced one [27]. The result of the filtered image after the Gaussian filter and the anisotropic diffusion filter is shown in the following figures.

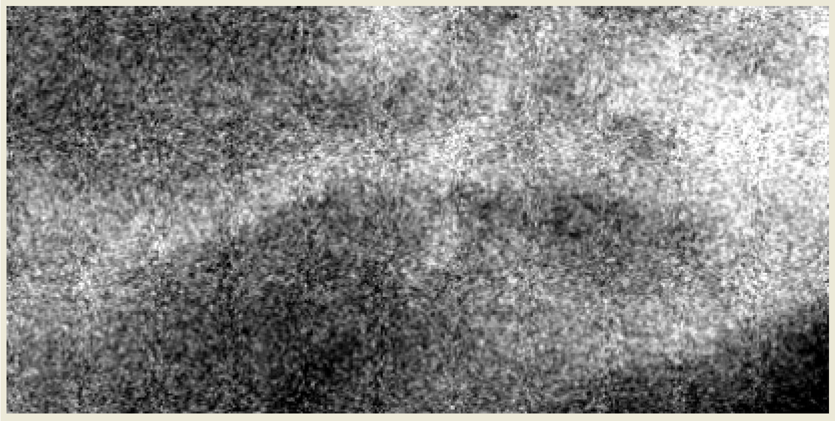

Figure 8: Original 400×200 region around ROI in a MVCBCT projection frame (Patient CBRT0003, frame 2)

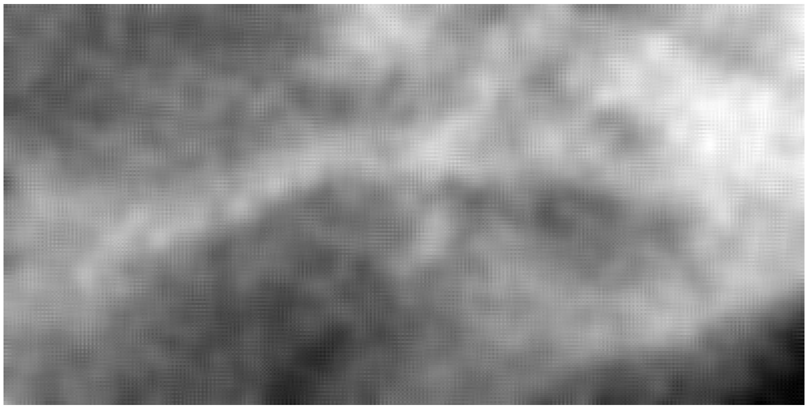

Figure 9: The region filtered by a 11×11 Gaussian smoothing filter. $\sigma = 5.0$

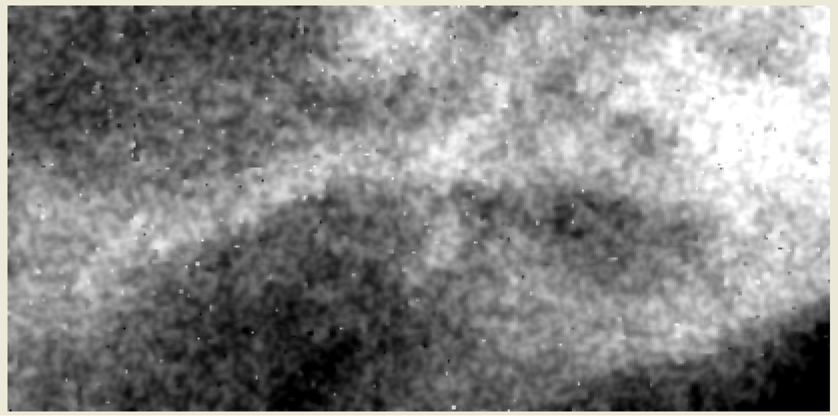

Figure 10: The region filtered by an anisotropic diffusion filter after 20 iterations. $K = 15$.

After filtering, the edge structure of the diaphragm is easier to identify using an edge extractor. The edge extraction technique applied in JED considers both the gradient amplitude and the gradient direction. A cost value is assigned for each pixel based on the gradient amplitude and direction, representing the probability of this pixel to be the diaphragm contour. First, the larger the gradient amplitude is, the more likely that this pixel is on an edge. Second, the closer the gradient direction is to the anticipated value, the more likely that this edge is a diaphragm contour. The cost function for each pixel can be expressed as:

$$g(x, y) = \left\|\nabla I(x, y)\right\| \cdot F(\arctan(\frac{\partial I / \partial y}{\partial I / \partial x}), x) \tag{2.12}$$

Where $g(x, y)$ is the cost value for pixel (x, y). In the following paragraphs, the expression (x, y) will be used instead of (R, z) (image detector coordinates) for clarity. $I(x, y)$ is the intensity for (x, y) and F is a function of the gradient direction angle and horizontal position. It returns a difference angle in radians between the gradient direction and an ideal direction at this horizontal position. The ideal direction is derived from the mean shape of a set of training samples of diaphragm contours. The size of the cost function region (CFR) is set at 400×200 typically. The calculation of the ideal direction and the illustration of training shapes will be given in the next section.

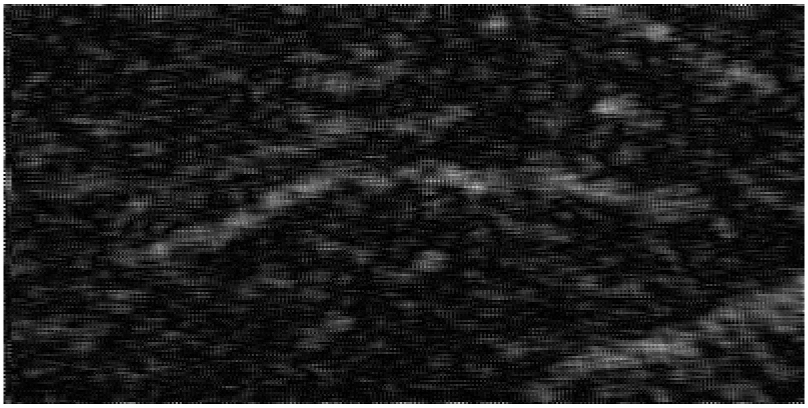

Figure 11: Cost image after Gaussian filter and edge extraction, CFR 400×200

Shape Analysis

With image filters and edge extraction techniques, the diaphragm contour can be well detected for high CNR frames. However, there is noise and interference that will make edge tracking process lost for these occasions. Prior shape information will provide additional knowledge that helps confine diaphragm tracking to the anticipated regions and restricts the wrong cases. The training data is generated from diaphragm shapes in 8 different patients. 5 frames which well represented the general pattern of the diaphragm are selected for each patient. Each shape contains 50 points equally distributed on the diaphragm contour, which are identified by a human on the JED platform. Each point in one shape corresponds to points in other shapes which have the same position on the diaphragm. The set of training shapes are shown as follows:

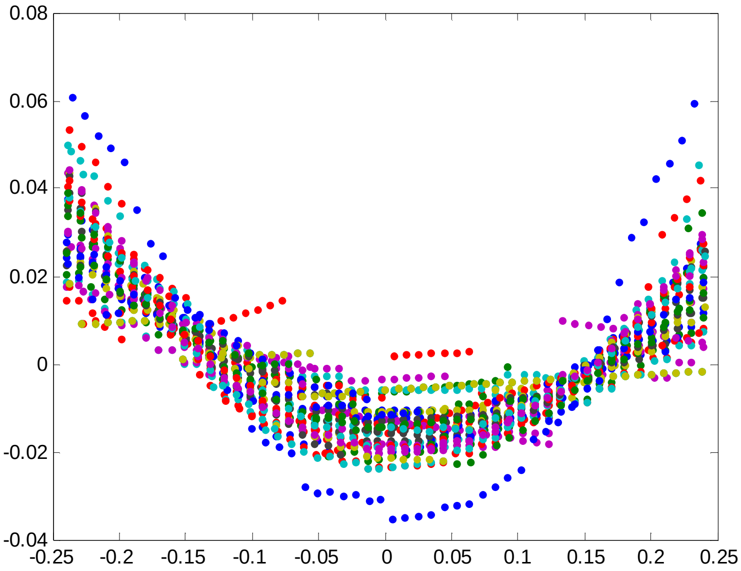

Figure 12: Uniformly scaled and aligned training shapes of diaphragm contours (the coordinates are aligned and inverted here compared to the pixel coordinate system)

And the mean shape of the training samples is also shown:

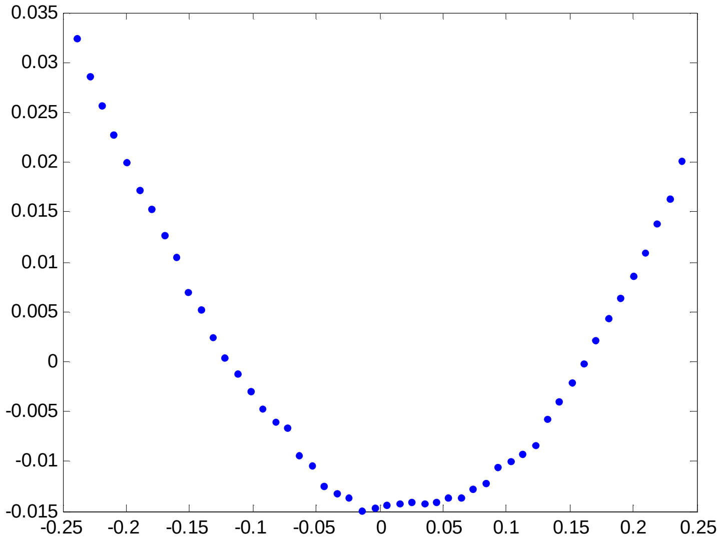

Figure 13: Uniformly scaled and aligned mean shape of training samples

The ideal direction for each point in the mean shape is calculated as the normal of this point, using the coordinates of two adjacent points. For a point (x_i, y_i), the normal can be calculated as:

$$\alpha_{ideal}(x_i) = -\frac{x_{i+1} - x_{i-1}}{y_{i+1} - y_{i-1}} \tag{2.13}$$

Note that after we get the training shape data, the scale and rotation angle of these shapes are not the same. In order to analyze the shape information, it is necessary to align and uniformly scale these shapes. The most common metric that measures the error of aligned shapes is the sum of distances:

$$D = \sum \left| x_i - \overline{x} \right|^2 \tag{2.14}$$

Where $\overline{x}$ is the mean shape of the training data and x_i is one certain shape. For shapes in 2D space, there are four parameters to be calculated to minimize the sum of the distances: the scale factor s, the rotation angle θ, and the translation factor x_{tran} and y_{tran}. The alignment follows the equation:

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_{tran} \\ y_{tran} \end{pmatrix} + s\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \tag{2.15}$$

There are several methods for calculating these parameters: calculating rotation angle after uniform scaling, calculating rotation angle and uniformed scale simultaneously, or projecting shape vectors into tangent space to avoid non-linearity [8]. The first method is applied in this work. In this method each shape is translated to the position where its centroid is at the origin. They are uniformly scaled to have the same norm. The rotation angle is calculated using a least-square method with singular value decomposition (SVD) [8].

The calculation of alignment uses an iterative algorithm illustrated on Page 14 in Cootes' book [7]. For each iteration, the mean shape is calculated. A reference shape is arbitrarily selected and the alignment is calculated for all the other shapes. The mean shape is updated in the next iteration. Typically, 2 to 3 iterations are enough to align the shapes into uniform positions. The aligned shapes of training samples are in figure 12.

As one diaphragm contour has 50 2D points, it can be considered as a point in R^n where $n = 100$. As the centroid, scale and the rotation angle of the shape is fixed, the shape vector has 96 free parameters. We wish to find the general pattern inside the training shapes. However, it is hard to identify the pattern in high dimensions. Principle Component Analysis (PCA) is a statistical technique that can reduce the number of dimensions of data without much loss of information, while highlighting the similarity and differences. Each shape can be expressed as:

$$s = (x_1, y_1, x_2, y_2, x_3, y_3, \ldots x_{50}, y_{50})^T \tag{2.16}$$

PCA is accomplished by calculating the eigenvectors and eigenvalues of the covariance matrix of the training samples. The covariance matrix is calculated from all the 40 training shapes. It has the expression as:

$$\mathrm{cov} = \frac{1}{n-1}(s_1, s_2, s_3, \ldots, s_{40})(s_1, s_2, s_3, \ldots, s_{40})^T \tag{2.17}$$

Note that all the shape vectors are aligned. In our case, cov is a 100×100 symmetric matrix. As the covariance matrix is semi positive definite, all the eigenvalues of cov are non-negative. In fact, each eigenvector of cov is one component of the data set. It represents a direction in 100 Dimension space. The eigenvalue represents the variance of data along the direction determined by the eigenvector. The eigenvector with the highest eigenvalue is the principle component. It is the most significant relationship between the data dimensions.

Now if the eigenvalues are sorted from highest to lowest, the components in order of significance are given. The process of reducing dimensions is to ignore the components of lesser significance, while keeping the more important ones. If the eigenvalues of ignored eigenvectors are small, the information is not lost too much. In our case, a few eigenvalues make up most of the variance of the data set.

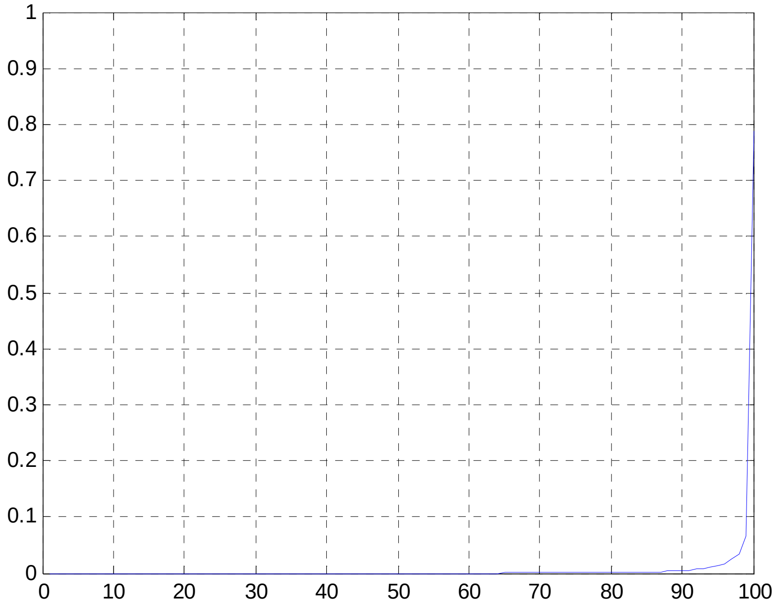

Figure 14: 100 eigenvalues of covariance matrix of training shapes. The horizontal axis is the number of eigenvalues sorted from lowest to highest, the vertical axis is the ratio of this eigenvalue to the sum of 100 eigenvalues.

In this figure, the largest eigenvalue takes up 78.9% of the sum of all the eigenvalues, with the second 6.64%, the third 3.49%, and the fourth 2.62%. In fact, the largest 12 eigenvalues take up 98.2% of the sum. We can ignore the other 88 components and just use these 12 eigenvectors to represent the data set without much loss of information. A linear model with 12 free parameters is established as:

$$x = \bar{x} + \Phi b \tag{2.18}$$

Where x is the shape generated by this linear model, which is a vector in 100 dimensional space. $\bar{x}$ is the mean shape. Φ is a matrix composed of the eigenvectors we select with the expression $\Phi = (v_1, v_2, v_3, ..., v_{12})$ and b is the parameter matrix for the 12 directions determined by 12 eigenvectors with $b = (b_1, b_2, b_3, ..., b_{12})^T$. This linear model cannot represent all the shapes in 100 dimensions. For an arbitrary shape, this linear model will find the shape closest to it by minimizing an error metric such as least squares. The parameters b will be calculated based on this metric for minimization. The advantage of prior shape information is that we can restrict parameters of b to confine the contour to be tracked in a reasonable shape. $|b_i| < 3\sqrt{\lambda_i}$ is applied in JED, where λ_i is the eigenvalue corresponding to this component.

Based on the established linear model of equation 2.18, ASM can be applied in diaphragm tracking. Given an initial instance of a diaphragm shape, ASM updates and confines the shape iteratively until convergence. During each iteration, for each point in a shape, ASM looks along a profile normal to the shape boundary through this point. $2l + 1$ pixels along the normal around this point are scanned in order to find the pixel with the minimum correlation (figure 15). For each pixel along the profile normal, the correlation is calculated as a Mahalanobis distance using the following equation:

$$f(g_s) = (g_s - \overline{g})^T S_g (g_s - \overline{g}) \qquad (2.19)$$

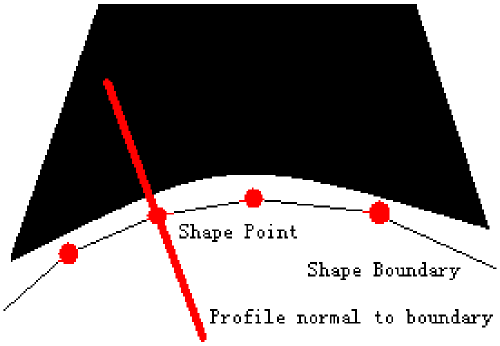

Figure 15: At each shape point, search along a profile normal to boundary

where g_s is a vector containing $2k+1$ elements. It represents the derivative of intensity in adjacent pixels along the normal direction. It scans for $2k+1$ adjacent pixels along the normal direction. $\overline{g}$ is the average derivative of intensities near this point in training samples and S_g is the covariance matrix of $\overline{g}$ of different shape points. The matching procedure is illustrated in the following figure:

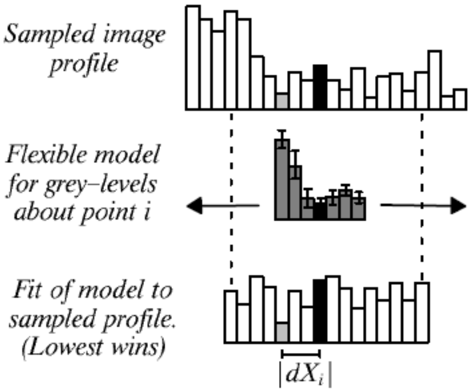
Sampled image profile
Flexible model for grey–levels about point i
Fit of model to sampled profile. (Lowest wins)
$|dX_i|$

Figure 16: Searching along the profile to find the best fit, $k = 2$ and $l = 6$

Image from Page 39 of [7], by permission from the author

Optimal Graph Search

Like ASM, many methods in video tracking use iterative schemes to find a local optimal solution. If the object function to be minimized is convex, the local optimum is then the global optimum. However, this is not always the case in real applications. Optimal graph search methods aim to find a surface in a 3D graph with a globally maximized sum of weights of nodes. As the projection data form a sequence of 2D images and can be considered as a video, we can use all 2D frames to construct a 3D graph where each vertex corresponds to one pixel in a certain frame with a weight calculated by equation 2.12. The mathematical expression of optimal graph search and the theoretic framework behind it can be seen in Wu and Chen's paper as the optimal V-weight net surface problem with smoothness constraints [36]. For completeness we will describe the concept as the following.

Given a 3D multi-column graph $G=(V,E)$ where V represents the set of vertices and E represents the set of edges of this graph, each vertex $v \in V$ having a real-valued weight $w(v)$. Find a net N in G such that the sum of the weight $\alpha(N)=\sum_{v\in V(N)} w(v)$ is minimized. The net N satisfies a smoothness constraint. For every two nodes $v_1(x_1,y_1,z_1), v_2(x_2,y_2,z_2) \in N$ in adjacent columns, i.e. $|x_1-x_2|+|y_1-y_2|=1$ where x_1,x_2,y_1,y_2 are integers which represent the position of the column in the base graph and z is the coordinate of a node in a column identified by coordinates (x,y), the difference of the position of the two nodes in each column should not be larger than a regulated constraint:

$$\begin{aligned} |z_1-z_2| &\le \Delta x, if\ |x_1-x_2|=1 \\ |z_1-z_2| &\le \Delta y, if\ |y_1-y_2|=1 \end{aligned} \tag{2.20}$$

where Δx and Δy are smoothness constraints along the x and y directions. The problem of single optimal surface detection is identical to the maximum closed set problem [15]. The maximum closed set problem is to find a closed subset of nodes $V^{'} \in V$ such that the sum of weights for this subset is maximized. The single optimal surface detection problem can be easily transformed into a maximum closed set problem without changing the graph. With $g(x, y, z)$ representing the original cost of the nodes, a new cost $c(x, y, z)$ for the maximum closed set can be calculated as:

$$c(x,y,z) = \begin{cases} g(x,y,z), z = 0 \\ g(x,y,z) - g(x,y,z-1), otherwise \end{cases} \qquad (2.21)$$

The maximum closed problem can be solved by a minimum cut algorithm on a related s,t graph [13]. This graph is denoted by G_{st} and it is derived by adding a source node s and a sink node t on the original graph G. All nodes with positive weight are connected from s with a capacity of the node weight in G, while all nodes with negative weights are connected to t with a capacity of the absolute value of the node weight in G. The arc capacity in the original graph G is set to infinity. It is proved that the maximum closed set S in G corresponds to the set S in the minimum cut of G_{st} [13].

There are many algorithms for the minimum cut problem. The running time of traditional algorithms such as the Ford-Fulkerson algorithm is $O(E \cdot f)$ where E is the number of edges in G_{st} and f is the maximum flow. The general push-relabel algorithm is $O(V^2E)$, where V is the number of nodes in G_{st}. Boykov and Kolmogorov proposed a method that is, at worst, $O(V^2EC)$ where C is the number of minimum cuts, but much faster in real applications [39]. Hochbaum proposed a pseudo-flow algorithm with complexities $O(VE \log V)$ [14]. In JED a former library of Boykov and Kolmogorov's algorithm is applied. The diaphragm contour in 200 frames can be detected simultaneously. The 400×200 CFR is resampled to a 30×30 region. The smoothness constraint of pixels in this resampled image is 1 for the x direction and 2 (equal to 13.3 pixels for original CFR pixels) for the t direction. (x, y, t) is used to represent coordinates instead of (x, y, z) to indicate that the 3D graph is an image sequence. Compared to the average velocity $4.44\,pixel\,/\,frame$ of the lung's motion, this intra-frame smoothness constraint is more appropriate considering velocity variations.

Dynamic Hough Transform

Optimal graph search establishes the intra-frame relationship by smoothness constraints. However, it does not account for the global shape pattern of the diaphragm contour. We would like to find a method that combines the prior shape knowledge and the intra-frame smoothness constraint. The previous methods on parametric models in video tracking are mainly focused on motion estimation, rather than shape matching [19, 22, 23]. A method developed by Baker and Barnes fits a parametric model onto a 3D Cochlea image [3], but no intra-frame constraint is applied in the registration process. Another method estimates the object motion by a 3D Hough transform of image motion, but this method can only detect constant motion [40]. A dynamic Hough transform was proposed that can detect arbitrary motion by detecting a trajectory in a Hough image [41, 42]. However this method assumes the shape of object is invariant throughout the frames. In this section a method based on the dynamic Hough transform that can detect a deformable object in arbitrary motion through an image sequence is described.

This method assumes that the diaphragm can be represented by a parabola with an axis parallel to the SI direction. The parabola can be described by a curvature parameter a and a vertex position (x_0, y_0). For each shape point $v_i \in V$ with horizontal coordinate x_i, where V is the point set of the parabola, the coordinates of v_i can be represented as:

$$v_i(x_i) = [x_i, a(x_i - x_0)^2 + y_0]^T \tag{2.22}$$

The goal is to choose appropriate parameters of a, x_0, y_0 to match this parabola template. For a given set of parameters of a, x_0, y_0, the metric function should be able to represent the propensity of this parabola to be the diaphragm contour. The matching metric function is designed to be the sum of the cost of all the pixels on the parabola:

$$M(a, x_0, y_0) = \sum_{x=1}^{w} g(v_i(x)) \tag{2.23}$$

Where w is the CFR width. For each frame, a reasonable range for each parameter of the parabola can be given: its vertex should be within the ROI, while the curvature parameter satisfies $0.0005 \le a \le 0.003$ based on experience. If the matching metric for every combination of parameters a, x_0, y_0 within the range is calculated, a 3D Hough image is generated such that every node in this graph (a, x_0, y_0) represents a parabola with a weight of metric $M(a, x_0, y_0)$. If 200 Hough images are stacked together from frame 1 to frame 200, a 4D metric graph is constructed. Each node is represented by (a, x_0, y_0, t) and has a weight $M(a, x_0, y_0, t)$, where t is the frame index.

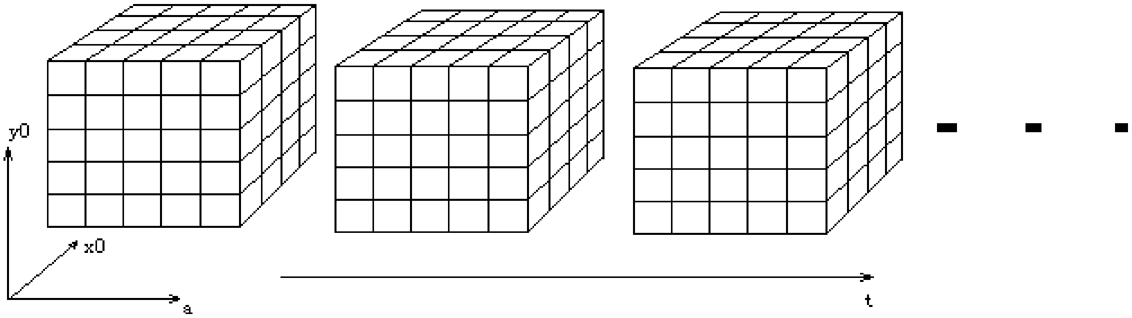

Figure 17: 3D Hough image sequence, each voxel has four coordinates (a, x_0, y_0, t)

With the 200 Hough images, the problem of diaphragm detection in 200 frames simultaneously becomes the problem of finding a voxel for each Hough image. We wish to find voxels with as large a metric value as possible. From ideas in optimal graph search, we can also incorporate intra-frame constraints. As there is only a small deformation of diaphragm contours between adjacent frames, the parameters (a, x_0, y_0) of the diaphragm parabola does not change too much. We can restrict the difference of parameters between adjacent frames to a reasonable limit. We can consider this problem as finding a continuous path traversing through a 4D graph composed of a 3D Hough image sequence. This path intersects with each 3D Hough image once with a maximum sum of weights. “Continuity” requirements of this path are satisfied by smoothness constraints. The mathematical statement is:

$$\max \sum_{t=1}^{200} M(a_t, x_{0t}, y_{0t}, t),\ \textit{subject to} \begin{cases} |a_t - a_{t-1}| \le \Delta a \\ |x_{0t} - x_{0(t-1)}| \le \Delta x_0 \\ |y_{0t} - y_{0(t-1)}| \le \Delta y_0 \end{cases} \qquad (2.24)$$

The method for calculation of the maximized fit is dynamic programming with path detection. Dynamic programming solves problems by decomposing them into overlapping sub-problems iteratively. It is much faster and uses less memory than naive methods such as exhaustive search. The 2D case for dynamic programming is well explained in Sonka’s book for applications of 2D image edge detection [33]. In this work the dynamic programming scheme is extended for path detection in a 3D Hough image sequence. The process of dynamic programming in a 3D image sequence is described by the following algorithm:

Given $M(a, x_0, y_0, t)$*:*

initiate a cumulative matrix $CM(a, x_0, y_0, t)$,

```
initiate three back tracking matrices $A(a, x_0, y_0, t)$, $X(a, x_0, y_0, t)$, $Y(a, x_0, y_0, t)$,
initiate three arrays $ar(200), xr(200), yr(200)$ for recording the result,
$CM(a, x_0, y_0, 1) = M(a, x_0, y_0, 1)$ for every $0.0005 \le a \le 0.003$, $(x_0, y_0)$ within ROI
For t=2 to 200
        $\Delta a', \Delta x_0', \Delta y_0' = \arg\max(CM(a+\Delta a', x_0+\Delta x_0', y_0+\Delta y_0', t-1) + M(a, x_0, y_0, t))$
        $s.t. |\Delta a'| \le |\Delta a|, |\Delta x_0'| \le |\Delta x_0|, |\Delta y_0'| \le |\Delta y_0|$
        $CM(a, x_0, y_0, t) = CM(a+\Delta a', x_0+\Delta x_0', y_0+\Delta y_0', t-1) + M(a, x_0, y_0, t)$
        $A(a, x_0, y_0, t) = \Delta a'$
        $X(a, x_0, y_0, t) = \Delta x_0'$
        $Y(a, x_0, y_0, t) = \Delta y_0'$
end (for).
$ar(200), xr(200), yr(200) = \arg\max(CM(ar(200), xr(200), yr(200), 200))$
For t=199 to 1
        $ar(t) = ar(t+1) - A(ar(t+1), xr(t+1), yr(t+1), t+1)$
        $xr(t) = xr(t+1) - X(ar(t+1), xr(t+1), yr(t+1), t+1)$
        $yr(t) = yr(t+1) - Y(ar(t+1), xr(t+1), yr(t+1), t+1)$
end (for).
```

The algorithm calculates the path with the maximum weight from the first Hough image to every voxel. The path for the voxel in the first Hough image is simply itself. For the other voxel, it looks around for each possible predecessor in the previous Hough image and chooses the voxel with the maximized weight. In this way it maintains a 4D array $CM(a, x_0, y_0, t)$ to record the cumulative weights. When the algorithm searches to the last Hough image, the path with the largest sum of weights can be easily found as the largest cumulative weight voxel in the last Hough image. The predecessor of this voxel along the path can easily be traced back to the first frame.

The algorithm scans each voxel and each of its predecessors one time. the total running time for this algorithm is $O(r(a)r(x_0)r(y_0)\Delta a\Delta x_0\Delta y_0)$ where $r(\cdot)$ means the number of possible values of the variable inside. For a, the value is sampled with a parameter grid spacing of 0.0003. 2 for y_0 and 5 for x_0. The smoothness constraint between adjacent frames is set at $15 pixel / frame$, which is much larger than the mean speed of diaphragm motion ($4.44 pixel / frame$) and close to the value used in the optimal graph search ($13.3 pixel / frame$).

CHAPTER THREE

RESULT AND DISCUSSION

Validation of Different Methods

In this work, different diaphragm tracking methods are tested, including boundary tracing as dynamic programming in 2D image(2D DP), ASM, optimal graph search (OPT), optimal graph search with edge weight on warped ROI (OPT-VCE) and dynamic Hough transform (DHT). To evaluate the accuracy of these tracking methods, we compared the diaphragm apex identified by a human expert against the diaphragm apex back projected into 3D room coordinates along the cranio-caudal axis. The error is calculated as the distance between the expert identified point and the algorithm generated point along the cranio-caudal axis. Only the cranio-caudal axis was tested because of the difficulty for the human expert to identify the horizontal coordinate of the diaphragm apex in views where the diaphragm is flat.

To get the position of the diaphragm apex in 3D coordinates, we have to first identify the diaphragm apex in 2D image planes. For DHT, the diaphragm apex can be easily derived as the parabola vertex (x_0, y_0). For other methods, 200 apex points are found simultaneously by searching for the maximum of the sum of heights of all the possible positions in 200 frames which satisfies the smoothness constraint. Dynamic programming is used in this searching process.

The second step is to estimate the position of the diaphragm apex in 3D. Unlike the ROI computation where the diaphragm apex at full exhale and full inhale can be determined by two user-identified frames each, the diaphragm apex in 3D has to be estimated from one frame. An interpolation method is used to calculate the (x_d, y_d) position of the diaphragm apex for an arbitrary frame, using the equation below:

$$x_d = x_{d1}\frac{z_{o2}-z_o}{z_{o2}-z_{o1}} + x_{d2}\frac{z_o - z_{o1}}{z_{o2}-z_{o1}}$$
$$y_d = y_{d1}\frac{z_{o2}-z_o}{z_{o2}-z_{o1}} + y_{d2}\frac{z_o - z_{o1}}{z_{o2}-z_{o1}} \qquad (3.1)$$

Where the subscripts 1 and 2 represent the coordinates of full exhale and full inhale points, respectively. Since there is little motion in x and y, the error of using this interpolation method is at most 1% [30]. With (x_d, y_d), z_d can be determined by calculating the similar triangle between the minified view and the image plane:

$$y_f = -x_d \sin\theta + y_d \cos\theta$$
$$z_d = \frac{SAD - y_f}{SAD} z_p \qquad (3.2)$$

The meaning of all the variables in 3.1 and 3.2 is the same for those in chapter 2. In this work the five methods are experimented on 21 patient data sets and the mean squared error for each patient image containing 200 frames are determined as in the following diagram:

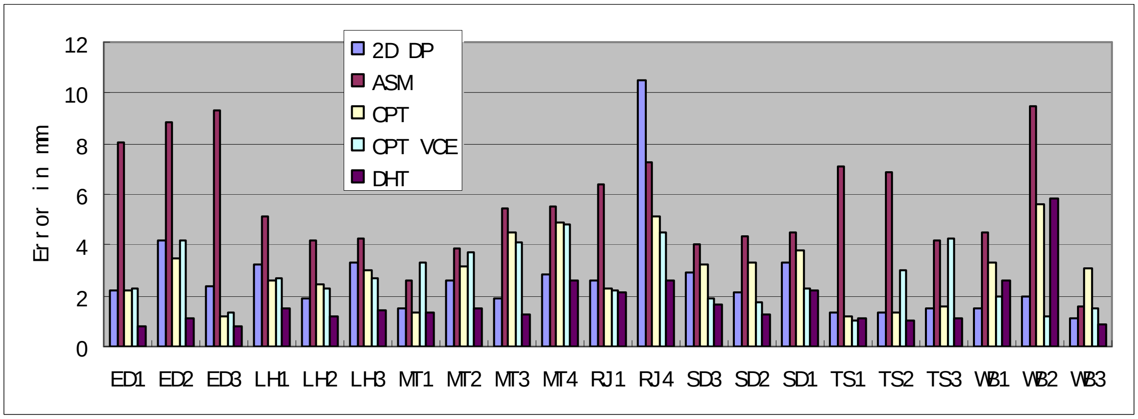

Figure 18: Mean squared error of 5 methods on 21 patient images

And the following diagram and table shows the mean and standard deviation of the error:

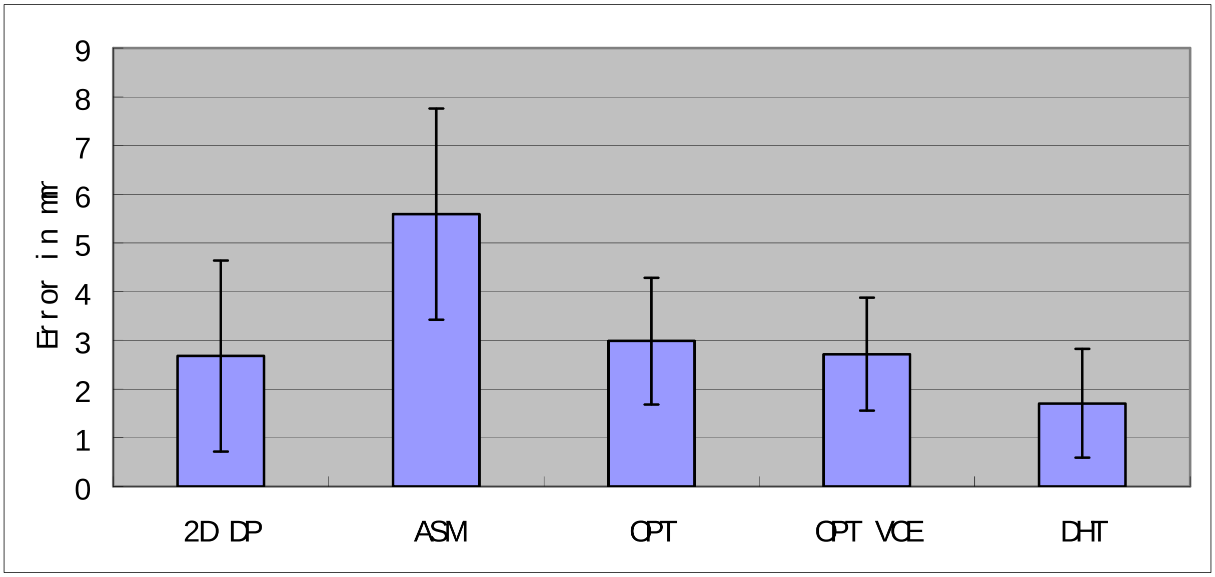

Figure 19: Mean and standard deviation of the error for 21 patient images

2D DP	ASM	OPT	OPT-VCE	DHT
2.680±1.963	5.589±2.170	2.986±1.299	2.716±1.156	1.707±1.117

Table 1: Mean and standard deviation of the error for 21 patient images

It can be seen that DHT performs better than the other techniques with a smaller mean and variance of error. 2D DP, OPT and OPT-VCE are less accurate than DHT but the error is stable. The error of ASM is relatively large as it loses track of the diaphragm in many frames. Some examples of the diaphragm contours detected by these methods are illustrated here:

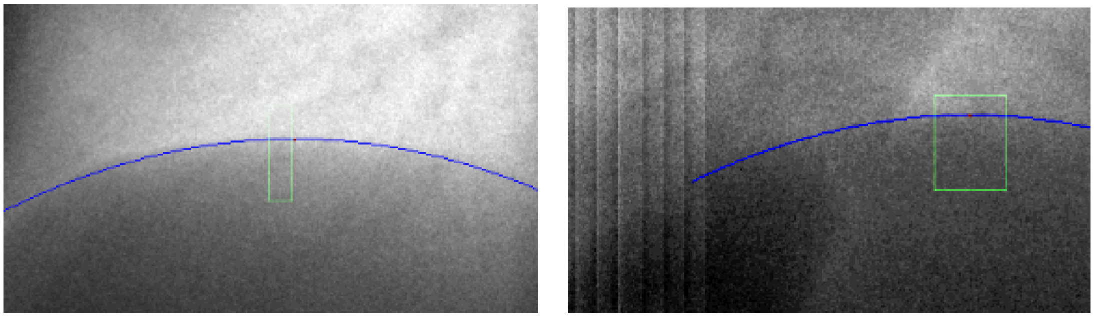

Figure 20: Diaphragm detected by DHT, a high contrast frame (left) and a low contrast frame with stripe artifact (right)

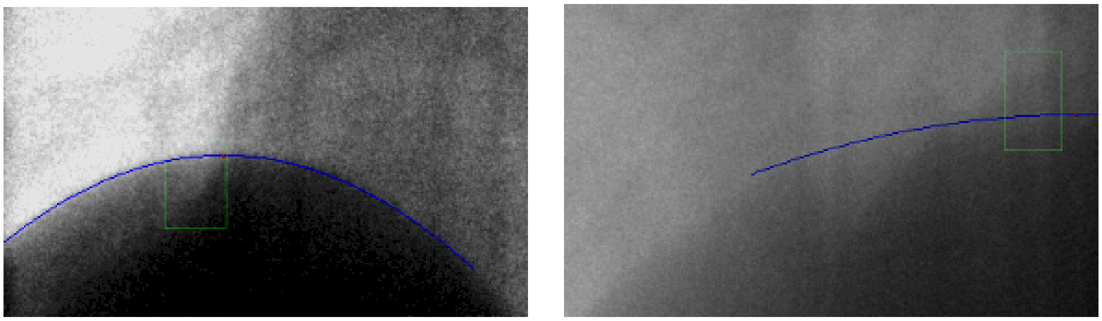

Figure 21: Diaphragm detected by DHT, a frame with confounding objects of heart and table (left) and a frame with overlapping lungs (right)

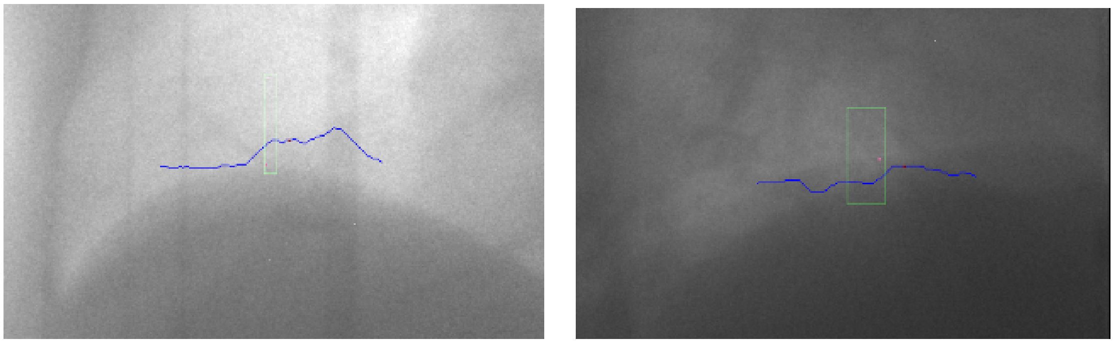

Figure 22: Diaphragm detected by 2D-DP:, a frame with interference of a bright area on the top (left) and a frame with overlapping lungs (right)

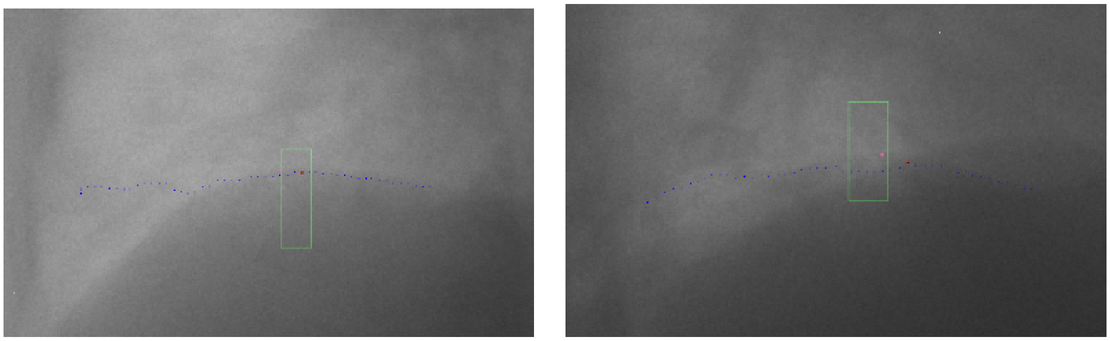

Figure 23: Diaphragm detected by ASM, a clear frame (left) and a frame with overlapping lungs (right)

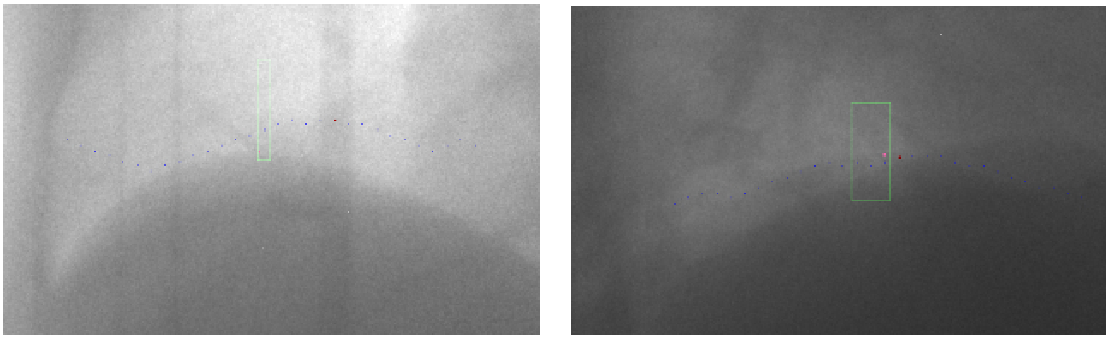

Figure 24: Diaphragm detected by OPT, a frame with interference of a bright area (left) and a frame with overlapping lungs (right)

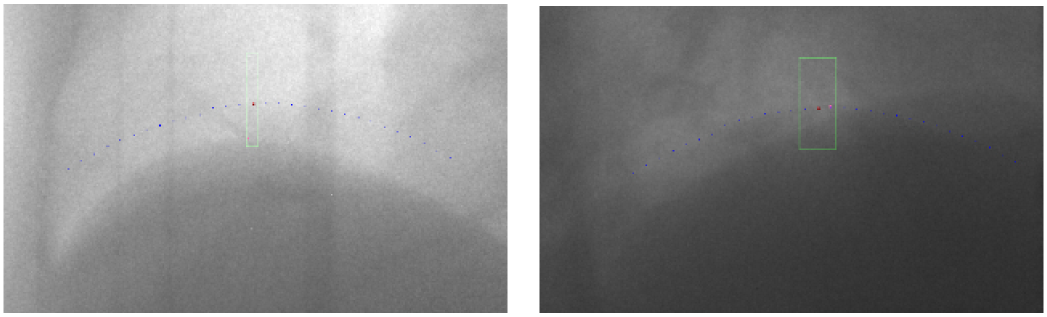

Figure 25: Diaphragm detected by OPT-VCE, a frame with interference of a bright area (left) and a frame with overlapping lungs (right)

It can be seen that an edge close to the diaphragm confuses the algorithms as it could be considered as the true diaphragm contour. Only DHT is not sensitive to local interference as it requires the parabola to be found, while the other algorithms can choose arbitrary paths for the contours. For the DHT algorithm, the use of the global minimum helps to distinguish between similar features. The parameter space for y_0 was put on a coarse grid at 2 pixel intervals, and at isocenter a pixel is about 0.27 mm, a systematic error of about 0.3 mm is present (this can be observed near the full exhale points at the bottom of the graph in figure 26).

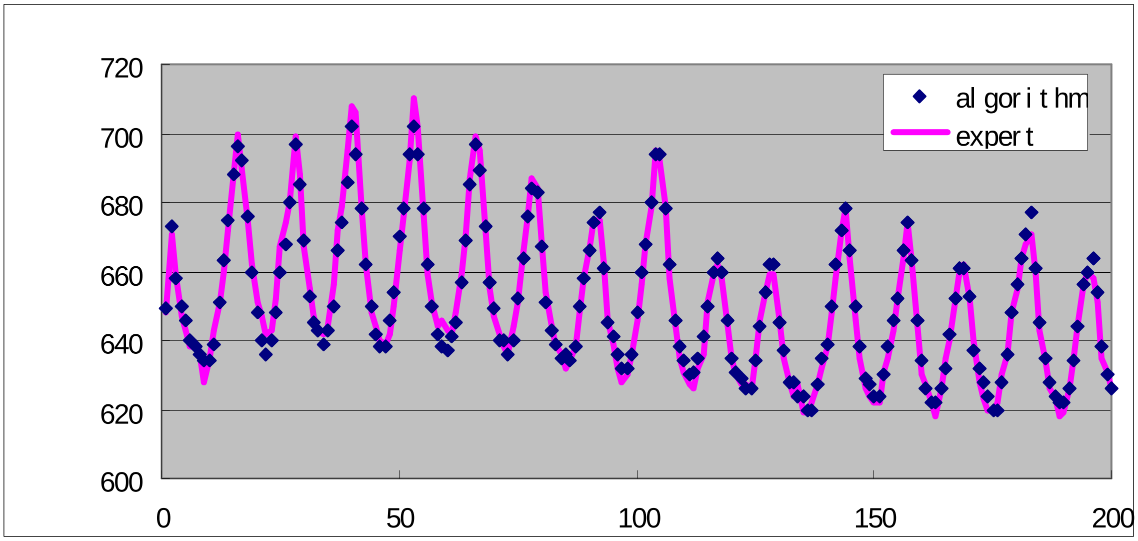

Figure 26: The vertical coordinate in the image plane plotted against frame number for the expert (pink line) and DHT algorithm (blue points).

Running Time

As the algorithms developed in this work will be implemented in clinical application, the running time is an important matter of concern. The diaphragm detection for 200 frames should be accomplished in less than one minute. Generally, the cost of less time is less accuracy. However, this is not always the case and with appropriate algorithm design and implementation, high accuracy and a reasonable running time can be achieved simultaneously. This section explores the running time in the methods applied in this work. The running time tested in this work is on a laptop with Intel dual-core T2300 1.66GHz CPU and 4G RAM in Microsoft Visual Basic 2008 debug mode with Windows XP.

The Gaussian filter is faster than anisotropic diffusion. Filtering a single frame takes about 2.90s for anisotropic diffusion with 20 iterations, while only 0.03s for an 11×11 Gaussian filter. The sum of time used in filtering 200 frames will be up to 10 minutes for anisotropic diffusion, and only 6s for the Gaussian filter. Considering the clinical requirement, the anisotropic diffusion filter is inappropriate. The Gaussian filter is applied in the preprocessing procedure.

The running time for computing the 400×200 CFR is about 0.1~0.15s for each frame. This filtering process is required by all the methods. The running time on 200 frames is about 20s, which should be taken into account.

The running time for the optimal graph search is about 7.5s for a 3D graph of $30\times30\times200$ nodes using Boykov and Kolmogorov's method [39]. Taking into account 20s for filtering, the total time should be 30s, while For ASM and 2D DP, the times are 48s and 42s, respectively.

For DHT, the running time of calculating the weights of nodes in the 3D graph of one frame is about 0.05s. The time for calculating 200 frames of the CFR and 3D graph takes a major part of the running time which takes 33s. The time for finding the maximum path is 19s. The total time required by DHT is 52s, which satisfies the clinical requirement, though it is slower than the other methods.

Note that the speed will be faster in a clinical application than the one tested on the laptop. One reason is the higher performance of the desktop computer used in the clinic. The second is that JED will be executed in application mode rather than debug mode, where the development environment performs other tasks. For example, the analysis of a CBCT by DHT can be done in about 38 seconds on a 2.66 GHz Intel quad-core 2 CPU, compared to 52s for the laptop.

Discussion and Future Work

DHT is the most accurate algorithm among the five methods tested in this work. However, several issues should be further considered to implement DHT in a real clinical application.

First, there is some human subjectivity for the expert identification in the cranio-caudal direction. The error between the algorithm and the human has two components: One is the systematic error generated by the algorithm, and one is caused by the variation of the human's subjectivity. In order to estimate this variation, a true gold standard needs to be established. The diaphragm apex should be identified by more human experts in order to get the difference between humans. By getting this variance, we can know more about to what extent the result achieved close to human identification.

For MVCBCT images, the motion speed is slower at full inhale and full exhale phases, while faster during the intermediate phases. A special case is that there is a large displacement between the first two frames because of the slower initial gantry rotation speed. However, the smoothness constraints applied in optimal graph search and DHT are all constant. These constraints are appropriate for the fast motion speed, but unnecessary for the slow one. An adaptive smoothness constraint is considered. In this scheme the algorithm may go through the 200 frames in two passes. The first pass is to get the phase information of the patient's respiration. Then an adaptive smoothness constraint can be determined by respiratory phase. The constraints can be set smaller for full inhale and full exhale phases, and larger for the intermediate phases. Then the algorithm will run a second time to locate the diaphragm accurately based on these constraints.

As a matter of fact, the parabola model can also be considered as a shape vector $s_{parabola} = (x_1, y_1, x_2, y_2, x_3, y_3, ...x_{50}, y_{50})^T$ in 100 dimensional space. But the parabola model cannot be represented by linear equation 2.18 because y_i is dependent on x_i. So instead of a line, the parabola model represents a surface in 100 dimensional space. It is hard to calculate the variance of training data to evaluate the performance of parabola model. From the observation of the tracking result, the parabola model matches most diaphragm contours well. But there are few cases where the shape of diaphragm is deformed due to pathological changes. More parameters should be considered to represent more various shapes of diaphragms for these cases.

We also notice that ASM performs poorly in this application. It loses track in many frames where the initial shape is too far away from the expected structure and the shape is trapped in a local extremum. An even worse case is that the diaphragm contour converges into a chunk, as each shape point searches for pixels along its normal direction. A multi resolution technique may resolve these problems. Using an image pyramid, the shape is first detected in a coarse image, then refined in a series of finer resolution images. This technique leads to a faster algorithm and is less likely to get stuck on the wrong image structure [7]. It can also be applied to algorithms of optimal graph search and dynamic programming.

CHAPTER FOUR

CONCLUSION

We have developed several algorithms for diaphragm detection in 2D views of cone-beam computed tomography (CBCT) raw data. These algorithms are tested on 21 Siemens megavoltage CBCT scans of lungs and the result is compared against the diaphragm apex identified by human experts. Among these algorithms dynamic Hough transform is sufficiently quick and accurate for motion determination prior to radiation therapy. The diaphragm was successfully detected in all 21 data sets, even for views with poor image quality and confounding objects. Each CBCT scan analysis (200 frames) took about 38 seconds on a 2.66 GHz Intel quad-core 2 CPU. The average cranio-caudal position error was 1.707 ± 1.117 mm. Other directions were not assessed due to uncertainties in expert identification.

REFERENCES

1. Ahn, S., Yi, B., Suh, Y., Kim, J., Lee, S., Shin, S., Shin, S. & Choi, E. A feasibility study on the prediction of tumour location in the lung from skin motion. *The British Journal of Radiology,* 77(919): 588, 2004.

2. Andrew, V. G., & Robert, E. T. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery,* 35(4), 1988.

3. Baker, G., & Barnes, N. Model-image registration of parametric shape models: Fitting a shell to the cochlea. *IJ - MICCAI Open-Source Workshop,* 2005

4. Barron, J. L., Fleet, D. J., Burkitt, T. A., & Beauchemin, S. S. Performance of optical flow techniques. *IJCV* 12(1): 43-77, 1992

5. Beuchemin, S. S., & Barron, J. L. 1. The computation of optical flow. *ACM Computing Surveys*, 27(3): 433, 1995.

6. Boyer, E. Object models from contour sequences. *Proc. 4th European Conference on Computer Vision (ECCV),* 2:109-118, 1996.

7. Cootes, T., Taylor, C., Lanitis, A., Active Shape Models: Evaluation of a Multi-Resolution Method for Improving Image Search, in Proc. *British Machine Vision Conference*, 327-336, 1994.

8. Cootes, T. F., Taylor, C. J., Graham, J., & Cooper, D. H. Training models of shape from sets of examples.*Proc. British Machine Vision Conference 1991,* 9-18. Published by Springer-Verlag, 1992

9. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. The ford-fulkerson method. *Introduction to algorithms,* 651. MIT Press and McGraw-Hill, 2001.

10. Davi, G., Alok, G., Luiz, A. C., & John, V. Dynamic programming for detecting, tracking, and matching deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(3): 294-302, 1995.

11. Frank, T., Nagel, H., Haag, M. 1., & Kollnig, H. 1. Tracking of occluded vehicles in traffic scenes. *Proc. 4th European Conference on Computer Visio (ECCV),* 2: 485-494, 1996.

12. Harrell, R. C., Slaughter, D. C. 1., & Adsit, P. D. 1. A fruit-tracking system for robotic harvesting. *Machine Vision and Applications*, 2(2): 69, 1989.

13. Hochbaum, D. A new-old algorithm for minimum-cut and maximum-flow in closure graphs. *Networks*, 37(4): 171, 2001.

14. Hochbaum, D. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research,* 56(4): 992, 2008.

15. Li, K., Wu, X., Chen, D., Sonka, M. Optimal surface segmentation in volumetric images-A graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 28(1): 119-134, 2006.

16. Kass, M., Witkin, A., & Terzopoulos, D. Snakes: active contour models. *International Journal of Computer Vision*, 1(4): 321-331, 1987

17. Keatley, E., Mageras, G., & Ling, C. (2000). Computer automated diaphragm motion quantification in a fluoroscopic movie. *Engineering inMedicine and Biology Society*, 3: 1749-1751. *Proc. 22nd annual international conference of the IEEE*, 2000.

18. Klim, S., Mortensen, S., Bodvarsson, B., Hyldstrup, L., & Thodberg, H. More active shape model. *Image and Vision Computing New Zealand*, 2003.

19. Duan, L., Wang, J., Zheng, Y., Xu, C., Tian, Q., Jin, J. & Lu, H. Shot-level camera motion estimation based on a parametric model. *TRECVID 2005*, Gaithersburg, Maryland, USA, 2005

20. Mageras, G. S., Yorke, E., Rosenzweig, K., Braban, L., Keatley, E., Ford, E., Leibel, S.A., Ling, C.C. Fluoroscopic evaluation of diaphragmatic motion reduction with a respiratory gated radiotherapy system. *Journal of Applied Clinical Medical Physics*, 2(4): 191-200, 2001.

21. Meyer, J., Richter, A., Baier, K., Wilbert, J., Guckenberger, M., & Flentje, M. Tracking moving objects with megavoltage portal imaging: A feasibility study. *Medical Physics*, 33(5): 1275-80, 2006.

22. Michael, J. B., & Yaser, Y. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. Proc. *5th International Conference on Computer Vision (ICCV)*, 374, 1995

23. Musse, O., Heitz, F., & Armspach, J. Topology preserving deformable image matching using constrained hierarchical parametric models. *IEEE Transactions on Image Processing*, 10(7):1081-1093, 2001

24. Nagel, H. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21(1): 85-117, 1983.

25. Morin, O. The development and role of megavoltage cone beam computed tomography in radiation oncology. Ph.D. dissertation, University of California, Berkeley, 2007.

26. Ozhasoglu, C., & Murphy, M. J. (2002). Issues in respiratory motion compensation during external-beam radiotherapy. *International Journal of Radiation Oncology, Biology, Physics*, 52(5): 1389-99, 2002.

27. Reitz, B., Gayou, O., Parda, D. S., & Miften, M. Monitoring tumor motion with on-line mega-voltage cone-beam computed tomography imaging in a cine mode. *Physics in Medicine Biology*, 53(4): 823-836, 2008.

28. Singh, A. (1990). An estimation-theoretic framework for image-flow computation. Proc. *3rd International Conference on Computer Vision (ICCV)*, 168-177, 1990

29. Siochi, R. Deriving motion from megavoltage localization cone beam computed tomography scans, *submitted to Physics in Medicine and Biology*.

30. Siochi, R. A projection point tracking method for gated 4DRT validation, MO-E-M100F-3, *49th AAPM Annual Meeting, Minneapolis, MN,* July 22 - July 26, 2007, (PowerPoint slides by private communication).

31. Siochi, R. PI, Verification of 4D radiation therapy, *NIH R21 grant proposal* 1R21CA133975-01. Submitted July 31, 2007.

32. Siochi, R. A projection point tracking method for gated 4DRT validation. *Medical Physics,* 34(6), 2528, 2007.

33. Sonka, M., Hlavac, V., & Boyle, R. Image processing, analysis, and machine vision, 3rd edition, Toronto: Thompson Learning, 2008.

34. TF, C., A, H., CJ, T., & J, H. Use of active shape models for locating structures in medical images. *Image and Vision Computing*, 12(6): 355, 1994.

35. Vedam, S. S., Kini, V. R., Keall, P. J., Ramakrishnan, V., Mostafavi, H., & Mohan, R. Quantifying the predictability of diaphragm motion during respiration with a noninvasive external marker. *Medical Physics*, 30(4): 505, 2003

36. Wu, X., Chen, D. Optimal net surface problems with applications. *Automata, languages and programming*, 775, 2002.

37. Xu, C., Prince, J. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359, 1998.

38. Xu, Q., Hamilton, R. J., Schowengerdt, R. A., Alexander, B., & Jiang, S. B. Lung tumor tracking in fluoroscopic video based on optical flow. *Medical Physics*, 35(12): 5351, 2008.

39. Boykov, Y. & Kolmogorov, V. An experimental comparison of min-Cut/Max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9): 1124-1137, 2004.

40. Iida, H., Hwang, J. & Ozawa, S. Extraction of motion from spatio-temporal image using the 3-dimensional Hough transform. *IEEE Pacific Rim Conference*. 1(19-21): 174-177, 1993.

41. Lappas, P., Carter, J.N. & Damper, R.I. Object tracking via the dynamic velocity Hough transform. *IEEE International Conference on Image Processing.* 2: 371-374, 2002.

42. Lappas, P., Carter, J.N. & Damper, R.I. Robust evidence-based object tracking. *Pattern Recognition Letters.* 23(1-3): 253-260, 2002.