A Vision System for Patient Positioning in Radiation Therapy

by

Salil Shukla

B.E., Maharashtra Institute of Technology, 1997

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Applied Science

in the Department of Mechanical Engineering

We accept this thesis as conforming

to the required standard

[redacted]

Dr. Colin Bradley, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. Zuoming Dong, Departmental Member (Dept. of Mechanical Engineering)

[redacted]

Dr. Harry Kwok, Outside Member (Dept. of Electrical and Computer Engineering)

[redacted]

Dr. Pan Agathoklis, External Examiner (Dept. of Electrical and Computer Engineering)

Supervisor: Dr. Colin Bradley

Abstract

This thesis outlines a new approach to position a patient on the treatment table for all the radiotherapy treatment sessions. Positioning is done by comparing the patient's contour from the diagnostic CT data to that from the Vision system. This vision system is accurate, reliable and easy to use.

The presence of cancerous cells is detected in the CT scan. Treatment planning is done based on the CT files. The patient undergoes several radiotherapy sessions that irradiate the cancerous tissues. The exact location of the patient during the diagnostic CT scan is known and used for dose calculations as well as positioning.

To ensure that the patient orientation is the same during each radiotherapy session, compensators or portal images are used. These procedures either expose the patient to more radiation or are tedious.

In this approach, the patient is positioned on the treatment table by a vision system. The vision system comprises a multiple line laser that projects equally spaced laser lines on the patient. A 2D image captured by the CCD camera is interpreted to generate the 3D contour of the patient. Various issues with processing the CT files and the 2D images are discussed. Comparison between the CT and the Vision contour is done in 3D by the Iterative Closest Point (ICP) algorithm. The iterative process calculates the overall positioning error that is used to reorient the treatment table.

Examiners:

[redacted]

Dr. Colin Bradley, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. Zuoming Dong, Departmental Member (Dept. of Mechanical Engineering)

[redacted]

Dr. Harry Kwok, Outside Member (Dept. of Electrical and Computer Engineering)

[redacted]

Dr. Pan Agathoklis, External Examiner (Dept. of Electrical and Computer Engineering)

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor, Dr. Colin Bradley for his help, guidance and support. He gave me every opportunity that would help me to excel throughout my MASc program.

Dr. Wayne Beckham and Dr. Derek Wells from the BC Cancer Agency were kind enough to give me the opportunity to work on this project. They carefully studied the results at every stage of the project and gave me suggestions for improvement.

All the members of the Optics lab at University of Victoria helped me to solve small problems I faced with developing algorithms, coding and writing reports. I sincerely thank everyone for their support, suggestions and thoughtful criticism.

Discussions with Dr. Harvey Richardson and Dr. Ron Podhorodeski were also valuable.

Last but not the least, I would like to thank all my friends for their encouragement and support.

Salil Shukla

Chapter 1 Introduction

1.1 Motivation

Medical imaging has become vital in the early detection and diagnosis of cancer. In many cases, it is possible to cure cancer completely by radiation therapy if detected at early stages. The therapy uses high-energy x-rays to stop cancer cells from dividing. X-rays deposit energy in the area being treated, damaging the genetic material of cells and making it impossible for these cells to divide. However, radiation therapy also damages normal cells in the vicinity of the cancerous cells. So it is very important to ensure that only the cancerous cells are being treated and not the healthy tissues. This work aims to minimize the destruction of the healthy cells by accurately positioning the patient.

The CT scan is the diagnostic tool that is done to detect the presence of tumors. If tumors are not controlled, they spread and lead to cancer. From the CT data, the exact location of the tumor with respect to the rotating CT tube or the machine coordinates can be computed and used for treatment planning or radiation dose calculations. However, radiation therapy can be given through different machines and is done over several sessions. Each time the patient lies down on the table for treatment, his/her exact position with respect to the machine is different. In this case, if accurate positioning is not done, the Planning Treatment Volume (PTV) is increased, thus posing more risk of destroying the healthy cells.

Conventional lead markers or tattoos are used to align the patient on the table. As nature's rule, every part of the human body undergoes some change in size and shape over time. In such a case, markers or tattoos are deflected and, therefore, are not helpful in registering the accurate position. Sometimes, thermoplastic compensators are used to register the patient's position. However, this procedure is cumbersome and always induces errors.

1.2 Literature Review

Much work has been done for precise patient positioning prior to treatment. "Conventional treatment setups attempt to re-create a particular patient orientation on the treatment table that is identical to that planned from a CT scan, normally using a few coplanar reference tattoo markers on the skin. Over the course of treatment, a patient's body surface undergoes changes virtually on a daily basis due to the unavoidable dynamics of internal/external forces. If the patient's body surface varies in shape, then it follows that the position of any external reference skin markers will also vary, leading to setup errors" [6].

Baroni [1], in his work, made use of multiple markers placed on patient skin for aligning and position monitoring. This was carried out by comparing current 3D positions of the markers, with those of an initial reference position acquired during the first irradiation session. Breathing phases were taken into account. It was possible to discriminate between different factors, which contributed to overall positioning error. However, this approach also required the body to be tattooed, which eventually added to the error.

Compensators are commonly used to align a patient. However, making and using the compensators involves many problems. Lerch [2] devised a method to manufacture paraffin-polyethylene tissue compensators by taking several photographs of the patient. However, the system suffered from the inherent delay in chemical processing, data transformation and machining.

Some physicians used Electronic Portal Images (EPI) to record the position of the patient. After the EPI were acquired, they were visually compared with the reference images. In the case of misalignment, the table was reoriented and the same procedure repeated until the position was fixed. This procedure was time consuming and not accurate because the images were compared by eye (physician's guess) in two dimensions.

In his approach, Cai [9] used the concept of fitting the sample points along the intended curve pair by taking the relative path length shift as an independent fitting parameter. This was done on the portal images, which was by default time consuming.

An improvement to the EPI approach was using live video of the patient for comparison with the reference image. A similar approach was used by Johnson [3] to align the patient and it was easier for the physician to interpret. Live video images were subtracted in real time from reference images, revealing the patient misalignment in multiple views. Immediate feedback was thus available to the physicians to reorient the patient. However, this method did not indicate the type of misalignment. Much time was often needed to orient the table based on judgments. Milliken [4] used a similar approach but gave more stress on the outline of ear and nose for registration, as these human parts do not change even when the facial expressions change. However, his method suffered from the same problems as Johnson.

In all the different positioning methods, lasers are initially used for lateral and longitudinal alignments. The use of lasers ensures that the patient is close to the reference position. Another approach was used by Van Lin [5] in his study. He considered the table-height set-up and laser alignment of skin marks for positioning. An offline correction protocol that reduced the subsequent set-up errors was used. The errors were minimized mostly in the ventro-dorsal direction. Again the comparisons were based on the skin marks that were used to check for the rotational errors along the longitudinal axis. Also, the offline correction protocol created an additional work load to the technician.

Two Orthographic x-rays were used by Schewe [6] to align the patient by matching the curves of the bony anatomy. Such images can be used to position the patient on a daily basis but the additional exposure to x-rays is undesirable.

Researchers used the planning and treatment CT data for patient positioning. Kreb [8] used a single CT slice to compare with the planning CT cube using an automated

positioning procedure. Several slice thicknesses and anatomical sites were used for comparison. However, comparison of the entire surface might give better results.

In Grahams [7] work on dynamic surface matching, the body surface height maps from the CT scan and optical sensor were compared. Color-coding was used to distinguish the surfaces that were compared in near real time. Simulated Annealing was used to find the best match which would consume more time due to the nature of the algorithm.

Thirion [10] proposed a new technique to perform 3D registration based on crest lines. The registration was done with CT or MRI images; however, the patient was exposed to x-rays each time for set-up.

An approach based on surface topography was presented by Berry [11]. With a multi-slit collimator, lines were projected on the patient and digitized and later compared to the reference images. This technique was good in a sense that for set-up, portal images nor CT scan were done and hence much time was saved. Also, the patient was exposed to fewer x-rays before treatment. In this approach the two images were compared to find the congruency (difference in the number of pixels) to compute the misalignment data which might not be accurate in comparison to a method comparing the contours.

1.3 Proposed Approach for Patient Positioning

The presence of a tumor is detected by the CT scan and the data files are used to generate a 3D contour of the patient. This contour information will be used for comparison with the contour generated by the vision system during each treatment session. To acquire the vision contour, a multiple line laser is projected on the patient and the lines are captured by a CCD camera. In this process, a 2D image is interpreted to give the 3D contour of the patient, by calculating the co-ordinates of each point lying on the laser line. The contours from both the processes are then compared to compute the misalignment and the same are shown graphically. The total transformation matrix generated from this registration is interpreted to find the rotation and translation values required to reorient the table. Time

consumed in the process is minimal (compared to conventional positioning techniques) and the technician has to spend less time in the radiation room.

A flowchart explaining the several steps in this approach is given in Appendix III. These steps are followed each time the patient visits the hospital for radiotherapy treatment.

Chapter 2 Experimental Set-up

The human body is made up of many cells that have the capability to reproduce when the body needs more cells. However, if the new cells are not required or consumed, they form an extra mass of tissues in the body. These tissues are tumors and are harmful if not removed. Some of the non-cancerous tumors do not spread to other parts of the body and are surgically removed. However, some tumors rapidly multiply or spread to other parts of the body and, if not irradiated, can pose a threat to life. These cancerous tumors can be treated if detected in early stages. Some non-surgical methods for removing the cancerous cells are chemotherapy, hormone therapy and radiation therapy. Radiation therapy is used when it is not possible to surgically remove the cancerous cells.

Computed Tomography (CT) imaging utilizes a rotating x-ray device to generate detailed cross sectional images or slices of nearly every part of the body. A fan beam of x-rays is produced as the frame holding the x-ray tube and the detector plate rotates around the patient. With one rotation of the frame, a slice is acquired. The slices are interpreted by the physician to locate the presence of cancerous cells. For any set-up, the slices are stored as separate files that can be interpreted for treatment planning and dosage delivery.

2.1 Obtaining the CT Data of the Bust

2.1.1 Making the Bust

As X-rays are harmful to healthy tissues, a live person could not be used to generate the sample CT data. A bust was used to generate the CT data instead. The bust shown in Figure 2.1 was made from UVEX material and the procedure is described below.

1. Liquid plaster is used to create the bust. Cream is applied to the patient's face and neck to prevent the plaster from sticking.
2. Liquid plaster is poured and allowed to dry and become hard.
3. The hardened plaster is used as a mold to generate the actual bust.

4. The UVEX plate is heated and placed on the mold in such a way that it conforms to the shape of the mold. On cooling, the shell is used as a bust for experimentation.

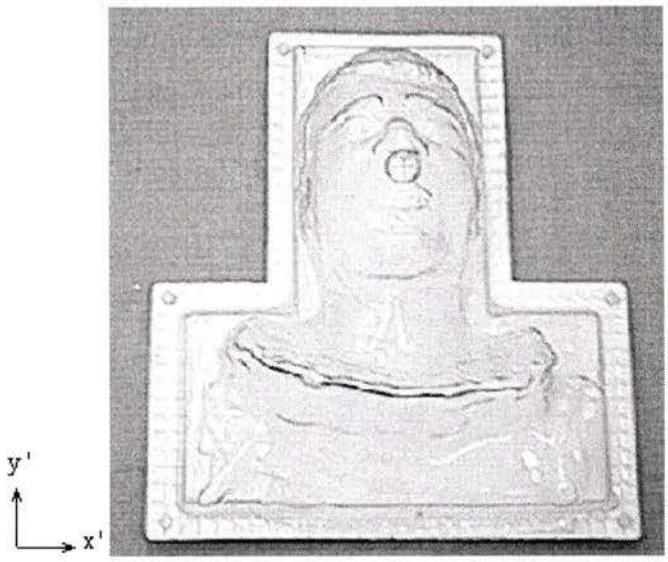


Figure 2.1: Transparent bust

In the vision system, the laser lines reflected poorly from the transparent surface of the bust, and therefore the bust was painted to make it opaque. Figure 2.2 below shows the bust after the coating process.

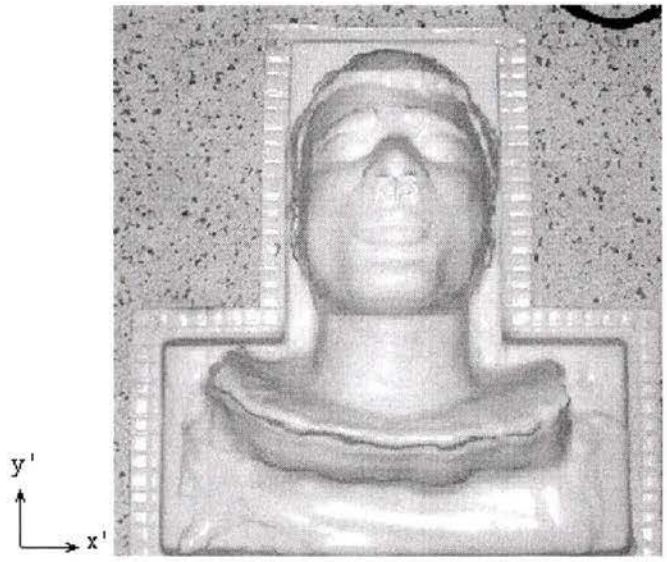


Figure 2.2: Opaque bust

2.1.2 Conventional CT Scan Set-up Procedure

Most of the hospitals use thermo-plastic compensators to register the patient's position. The compensator is essentially a plastic sheet with hard boundary and locating pins. The treatment table has special grooves to accommodate the compensator's locating pins. During subsequent set-ups, the compensator is placed on the patient's face and fixed to the table with the help of the locating pins. Stickers (with lead crosshairs) are stuck on the compensator for lateral and longitudinal alignment. Laser cross beams are used to align along the two directions. If treating for head-neck cancer, the patient is moved towards the center of the rotating tube, aligned with lasers and lead stickers and this corresponds to the machine's origin (0, 0, 0) position for treatment.

2.1.3 CT Scan of the Bust

Since the bust was rigid and stationary, none of the immobilization devices were required. Stickers were put in three different positions on the face, so that the bust was aligned along the lateral and longitudinal directions. The CT scan was made with an inter-slice distance of 3 mm. A total of 91 slices were acquired and then stored at a known location on the computer which was linked to the CT scanner.

For some slices, the contours were disjoint. To rectify the disjointed contours, in-house software was used. Once this was done, the slice data files were ready for interpretation. The slices all put together, generate the 3D contour of the patient which is shown in Figure 2.3. In bust in the figure is rotated so to see the contour properly.

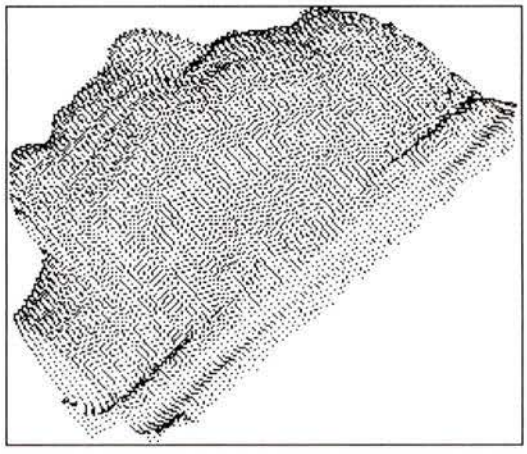


Figure 2.3: 3D contour of the bust

2.2 Set- up of the Vision System

The CT scan was done on a machine at the BC Cancer Agency, Victoria. To simulate the treatment table, a test bed was built at the Optics laboratory in University of Victoria. A frame holding the camera, lens and the laser projector was made from T-slotted aluminum extrusions (beams). The position of the camera on the frame could be adjusted, making the test-bed more suitable for experimentation. The height of the vertical beams was 8 feet. The requirement at the hospital was a high frame that would not block the way of the regular set-up devices. The camera & lens assembly were mounted at an angle of 17° with respect to the vertical axis. A multi-line laser projected lines on the patient which were captured by the camera. The details of the CCD camera, lens and the laser projector can be found in Appendix I, an image of the test rig in Appendix II.

2.3 Laser Triangulation

In a laser triangular system, a laser spot or line is projected on the surface to be measured and is calibrated against the known distances or angles. Such systems are used to record the distance or generate profiles of the objects for analysis. Laser triangulation is used in many machine vision applications.

2.3.1 Point Laser Triangulation

In point-laser triangulation, a laser point is projected on the target and an image is recorded with the CCD camera. The point laser triangulation set-up is shown in Figure 2.4 below. From the known parameters, the distance to the target can be calculated. This type of system can be used to record the 3D profile of an object by giving fixed translations (along 2 axes) to the laser and recording the points. However, the process can take a long time to digitize the complete surface. The data generated is unstructured.

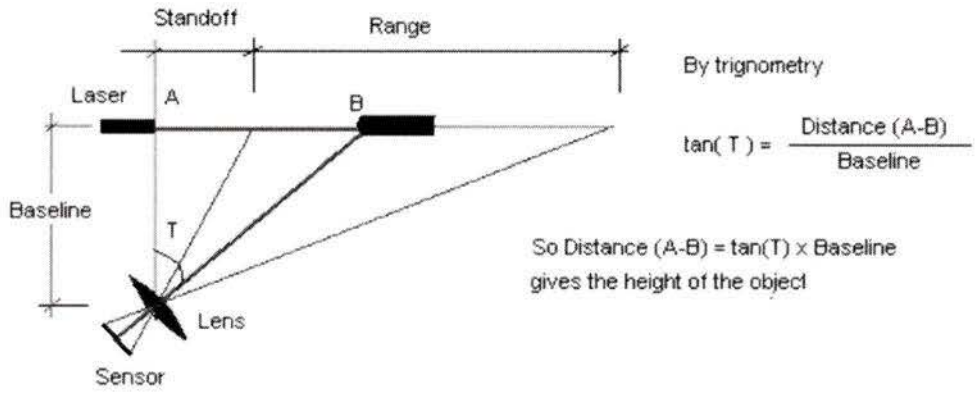


Figure 2.4: Point laser triangulation

2.3.2 Stripe Laser Triangulation

A stripe laser triangulation system projects a line on the target which is captured by the CCD camera. The set-up is shown in Figure 2.5 below. However to generate the contour of the entire target, an additional translation device is required to move the laser along one direction. The data generated by the stripe laser triangulation system is structured data.

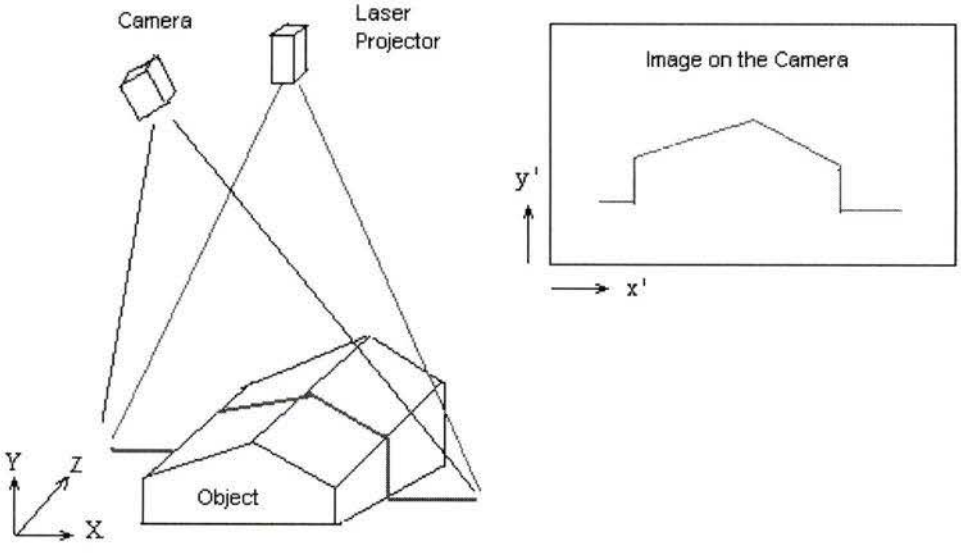


Figure 2.5: Single stripe laser triangulation

2.3.3 Multiple Stripe Laser Triangulation

Point-laser and stripe-laser triangulation systems each have their advantages. However, they cannot generate the complete contour of an object without the help of a translation device. The complete 3D contour can be generated by using a single laser projector that generates multiple lines. The projector has a detachable head that is fixed to the laser and can generate profiles such as dots, concentric circles, multiple lines and multiple dots. Figure 2.6 below shows the multiple stripe laser projecting 3 lines on the object.

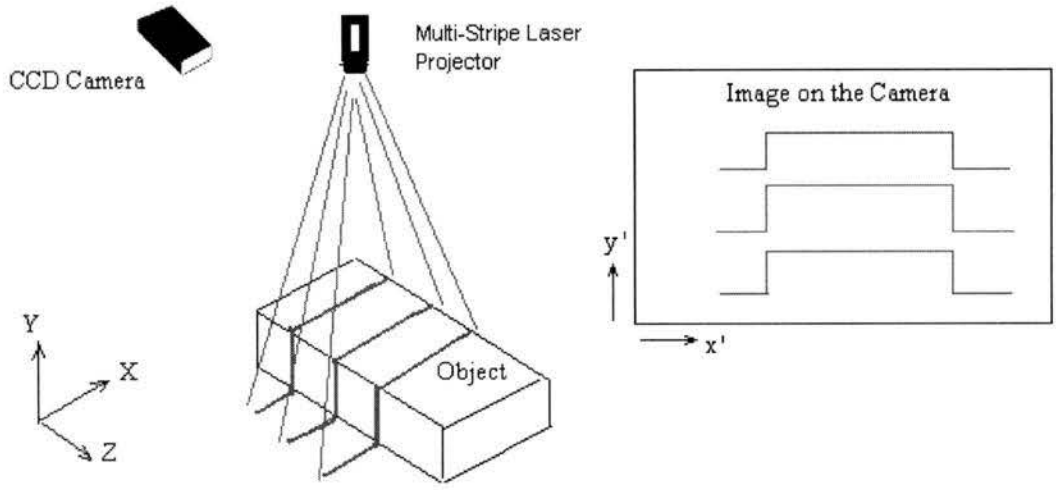


Figure 2.6: Multiple stripe laser triangulation

A single 2D image of the multiple laser lines taken by the CCD camera can be used to generate a 3D profile of the object. The distance between the laser projector and the CCD camera is fixed and the angle between the individual laser lines is known. With suitable calibration procedures, one can find the exact coordinates of each point in the image..

However, care must be taken while generating a 3D contour from the 2D image. If the angle between the laser lines is small, different algorithms must be implemented to accurately separate the laser lines that overlap. Also, to find the exact coordinate values of the data points, calibration of individual lines is required since their distance from the camera varies. These details are covered in Chapter 5.

Chapter 3 Processing the CT Data

The CT scan is done on every patient to check the presence of cancerous cells. When the patient is aligned on the table with the help of lasers and lead stickers, the position corresponds to Z = 0. The table is moved backwards and the x-rays are passed through the rotation tube at predetermined intervals. On one rotation of the tube, a single slice is acquired. With the movement of the table along Z axis and simultaneously passing x-rays, the entire part of interest is scanned. Each slice generates a file which has information on the slice thickness, X & Y contour values, Z value, scaling factors and the patient details.

A 3D contour is generated by decoding each of the slice files with an appropriate scaling factor, and combining them together for user visualization. However, the raw data needs much preprocessing to be interpreted and used.

A 3D model in Stereolithography (STL) format is generated that can be used for different applications. These applications include treatment planning, patient positioning on the table, the manufacture of a prototype model for simulation purposes, and a compensator for "fixing" the patient to the table. The various algorithms used in this process are discussed below.

3.1 Data Extraction

Figure 3.1 below displays the original 3D point data of the bust resembling a head-neck cancer patient. The cloud of data points were extracted from a total of 91 CT slices, with an inter-slice distance of 3.00 mm. Each slice was stored with a specific name and extension, for example "abcdefghi.10", "abcdefghi.11", "abcdefghi.12". Software was written to extract contour points from each slice and the points were displayed with appropriate scaling factors.

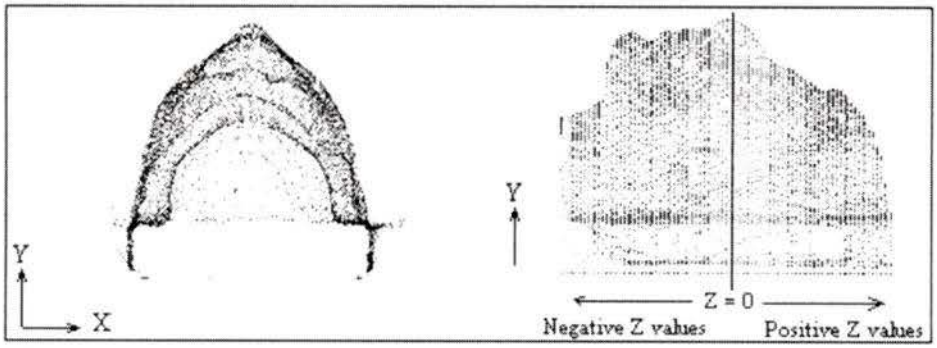


Figure 3.1: Original CT data

3.2 Preprocessing

The distance between the points in any slices was as low as 1.00mm and high as 20.00 mm. If the distance between the points was low, the computation time increased because there were more calculations, whereas if the distance was high, it did not give an exact idea of the shape. Preprocessing was done to counter this problem and the gaps between the points were filled by linearly interpolating points. The new dense but regularized data is shown in Figure 3.2

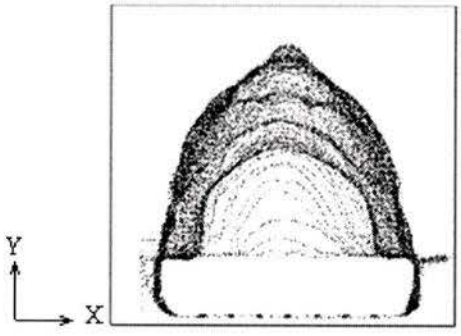


Figure 3.2: Preprocessed CT data

The algorithm used in pre processing is explained below:

for each slice

start from the first point

find the distance between the first point and second point

$$dist = \left\| \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2} \right\|$$

if distance < 3 mm, Goto the next point
if 3 mm < distance < 6 mm, split the points once
if 6 mm < distance < 12 mm, split the points twice
if distance > 12 mm, split the points thrice

next point

next slice

3.3 Selection of the Contour Data

The bust had a 55 mm-high base to keep it stable when resting on a flat surface. The contour points pertaining to this base were recorded in the scan. The unwanted points pertaining to the base were not needed and an option was provided to remove these points. These points will not show up in the CT scan of a patient. The data points were arranged in such a way that the starting point for each slice was on the same plane. Because of this rearrangement, the meshing algorithm generated smooth triangles. Figure 3.3 shows the 3D data with the bottom points removed.

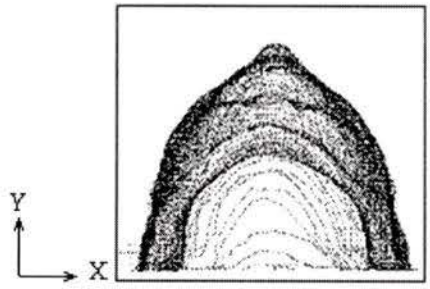


Figure 3.3: Removing the bottom part of the bust

A Y value was selected by the user in the software for a cut-off to enable removal of the base points. The algorithm implemented is explained below

```
for each slice
        for each point (X, Y, Z)
                if Y > user selected value Then
                        retain the point
                else
                        delete the point
                endif
        next point
next slice
```

The points were sorted in such a way that the first point on every slice had a Y value either equal or close to the user selected value. Virtually all the points were close to each other and this proximity made the meshing algorithm generate uniform triangles. When the CT scan is done on a patient, this step is unnecessary. For such scanned data, the points are arranged at a later stage in the data ordering section.

3.4 Data Thinning

At the preprocessing stage, an attempt was made to keep the distance between the points equal to the slice distance. However, if this distance was less than the slice distance, the points were deleted as it made the cloud data dense. The method by Lee [13] reduces the cloud of points using 3D grids. For this application, by data thinning, the distance between the points was controlled and maintained to be equal to the slice distance. This arrangement was helpful for meshing, as the triangles generated were uniform. Also uniformity in the data improved the comparison with the vision contour. Figure 3.4 shows the same part with two different inter-point distances.

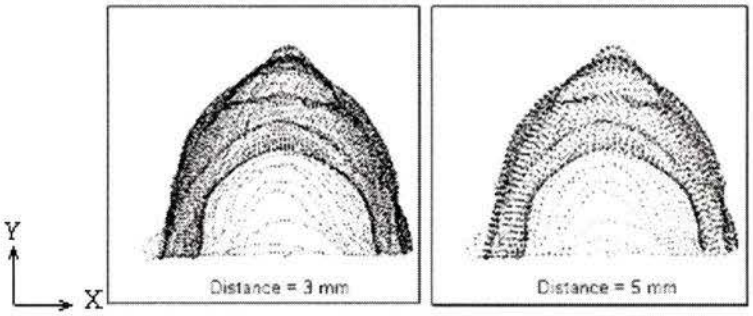


Figure 3.4: Data thinning with different inter-point distances

The algorithm is explained below

```
for each slice
        for each point (X,Y,Z)
                if distance is in between slice distance +/- 0.1, retain the point
                        if distance < slice distance – 0.1, delete the point
                        if distance > slice distance + 0.1, interpolate point
                endif
        next point
next slice
```

3.5 Data Ordering

When the data points were extracted from the CT slices, it was observed that the starting points for every slice were randomly oriented along the slice as shown in Figure 3.5. Data ordering for the bust was done (see section 3.3). However for the actual patient CT data, this option would not be used and therefore the data needed ordering. With the ordered

data, the starting points of the slices were close, meshing algorithm became easier to implement, and the triangles formed were uniform.

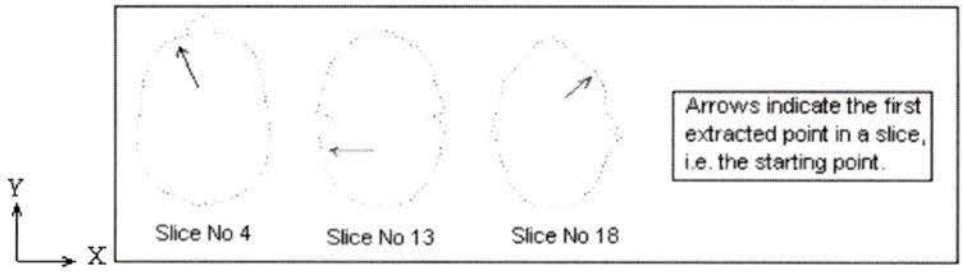


Figure 3.5: Different slices with starting points starting on different planes.

Ordering the data was done in 4 steps which are explained below:

Step I: The distance of the first point (first slice) from each point on every other slice was calculated and stored in an array.

Step II: The smallest distance for each slice and the number of points in each slice were calculated and stored. As shown below in Figure 3.6, the slices 1, 2, 3 & 4 are equidistant. The distance of any point between the slices along the line AB will be the shortest distance from the first point of slice 1. Therefore, the first point on slice 1 was selected and compared to all the other points on various slices. All points lying on this line are chosen as the starting points for every slice.

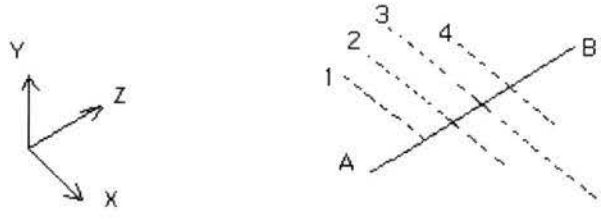


Figure 3.6: Finding the smallest distance point

The algorithm is explained below

```
for each slice
        for each point(X,Y,Z)
                if distance(slice, point) <> 0# Then
                        if low <= distance (slice, point) Then
                                low = low and Lowno = Lowno
                        else
```

```
                        low = distance(slice, point) and Lowno = point
                    end If
                        iter = iter + 1
                end If
            next point
                    smallest_dist_position(slice) = Lowno
                    max_no_points(slice) = iter
        next slice
```

Step III: The points are separated depending on the smallest distance point. The smallest distance point can be any numbered point on the other slice and was considered as the separation point. Figure 3.7 shows three of the slices separated by the smallest distance point.

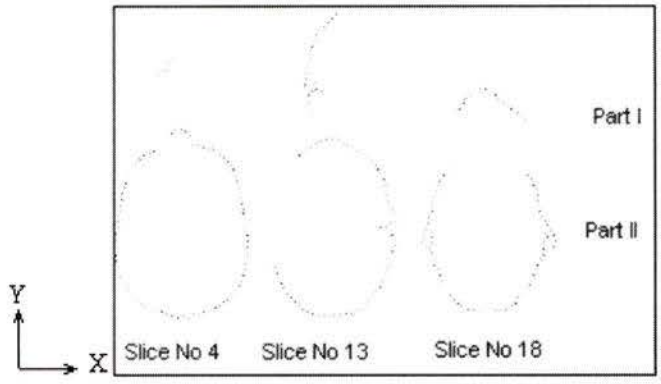


Figure 3.7: Separating the two parts of a single slice

The algorithm is explained below

```
'Put one part of the data points in one array
   for each slice
      for ptno = 1 To smallest_dist_position (slice)
         bthin(1, slice, bt) = points(1, slice, ptno)        'bthin() = one array
         bthin(2, slice, bt) = points(2, slice, ptno         'points()= array of pts
         bthin(3, slice, bt) = points(3, slice, ptno)
         bt = bt + 1
      next ptno
   next slice

   'Put the second part of the data points in another array
   for each slice
      for ptno = smallest_dist_position (slice) To max_no_points (slice)
         athin(1, slice, at) = points(1, slice, ptno)        'athin()= second array
         athin(2, slice, at) = points(2, slice, ptno)
         athin(3, slice, at) = points(3, slice, ptno)
         at = at + 1
      next ptno
         at = 1
   next slice
```

Step IV: Once the points were separated (depending on the smallest distance point), they were merged together in one array. The starting points for every slice formed by the new array lie very close as shown in Figure 3.8.

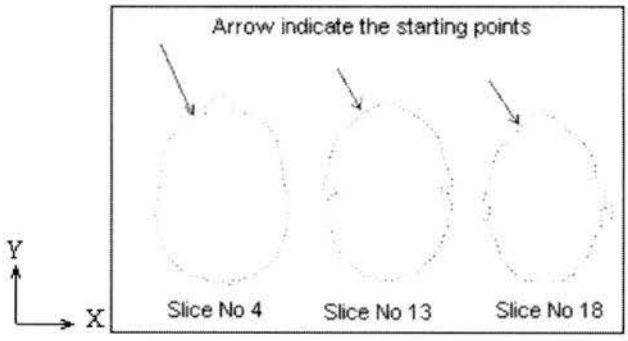


Figure 3.8: Fixing the starting point on the same plane

Merging algorithm is explained below:

```
'Put the points together - First part
   for each slice
      for each point(X,Y,Z)
                if athin(1, slice, point) <> 0# And athin(2, slice, point) <> 0# Then
                      put the points in the array abthin( ) from athin( )
               end If
      next point
   next slice

   'Put the points together - Second part
   for each slice
      for each point(X,Y,Z)
         if abthin(1, slice, point) = 0# And abthin(2, slice, point) = 0# Then
               Put the points in array abthin( ) from bthin( )
         end If
      next ptno
   next slice
```

3.6 Generating a surface

Several types of algorithms are available for generating meshes from a set of points. A single NURB patch is not sufficient to define the human shape and so several NURB patches are required. This makes the procedure complex and time consuming. The same problem is faced with an algorithm using Bezier curves.

According to Bradley [12], there are several advantages of using triangular meshing techniques over NURB or Bezier surfaces. Triangular meshes can represent general shapes and do not require manual segmentation. For structured data, triangular meshing is effective and easy to implement because the interval between the scan lines is defined. There are many different approaches for creating the triangular meshes. The one used here was a simple algorithm and it joined two points on one slice and one on the other as shown in Figure 3.9. This approach was only possible if the starting points for every slice were close to each other. Data ordering done previously was therefore an important step to enable implementation of this algorithm.

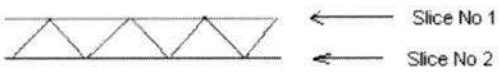


Figure 3.9: Simple approach to create triangles

To check the accuracy and effectiveness of the meshing algorithm, a STL file was created. The STL format is an ASCII or binary file used in manufacturing for rapid prototyping applications. For every triangle, the STL format stored three points forming the triangle and their normals in a specific format. The file can be opened in any CAD package and rendered.

A typical STL file format is explained below

```
Open "c:\stlfile1.stl" For Output As #1
print #1, "solid OBJECT"
for each slice
        for point  = 1 To iterno(slice)
                print #1, "facet normal" & " " & NX & " " & NY & " " & NZ
                print #1, "outer loop"
                print #1, "vertex" & " " & X1 & " " & Y1 & " " & Z1
                print #1, "vertex" & " " & X2 & " " & Y2 & " " & Z2
                print #1, "vertex" & " " & X3 & " " & Y3 & " " & Z3
                print #1, "endloop"
                print #1, "endfacet"
        next point
        next slice
print #1, "endsolid OBJECT"
close #1
```

A 3D model (Figure 3.10) of the bust was generated using the different inter-point distances. Table 1 below lists the details:

Model No	1	2	3
Inter-point dist	3 mm	8 mm	15 mm
No. of vertices	51642	19008	10302
No of triangles	17214	6336	3434

Table 3.1: Variation in inter point distances

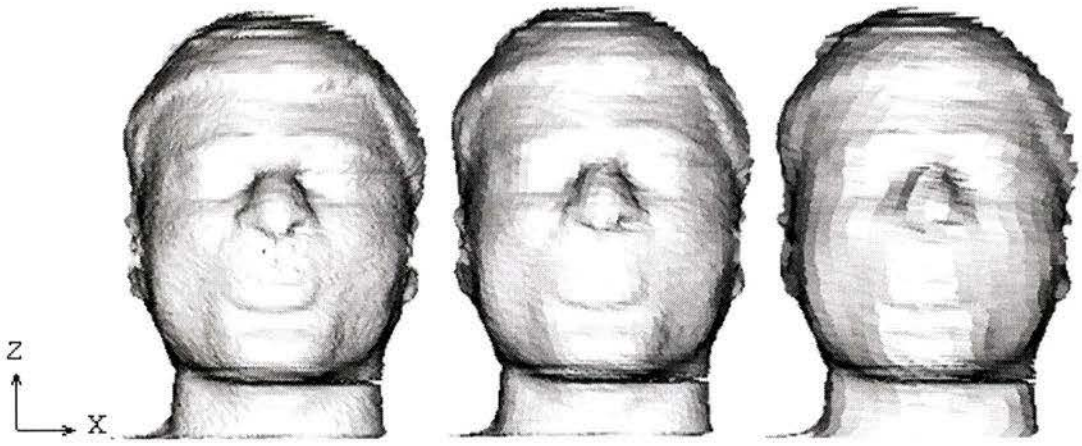


Figure 3.10: 3D rendering of the bust with different inter-point distances

For the first model, the inter-point distance within any slice was equal to the inter-slice distance. High level of detail was available in the first model. The small closely spaced triangles in model 1 have five times more details than the large ones in model 3.

Using the STL file, a compensator can be manufactured which can be used to align the patient during the daily setups. The STL file can also be used to make a 3D model for simulation purposes.

For accurate patient positioning, this 3D surface can be used effectively for comparison, if the vision data is similar in terms of the distance between the slices. If the vision data is not dense, then a surface cannot be generated and in such a case, the best approach is to compare the two 3D point sets and not the surfaces.

Chapter 4 Image Distortion, Compensation and Calibration

Any digital image is considered to be ideal when it is free from distortion. A major component of the system which distorts the image is the lens. Customized lenses have comparatively less distortion but there is always a tradeoff between the cost of the lens and distortion free image producing capability of the lens. Telecentric lenses give good results and have an advantage over zoom lenses because the images produced are not distorted, but these lenses are expensive and have a short working range.

4.1 Lens and Laser Line Distortion and Correction

Coma, astigmatism, spherical distortion, barrel distortion and pincushion distortion are some of the several distortions encountered while processing digital images. In barrel distortion (Fig 4.1), the outer points tend to crowd together; whereas in pincushion distortion (Fig 4.2), the outer points spread outwards. These types of distortions are mainly caused by the improper curvature of the lens. A combination of barrel and pincushion distortion can also be present in a system and is comparatively difficult to correct.

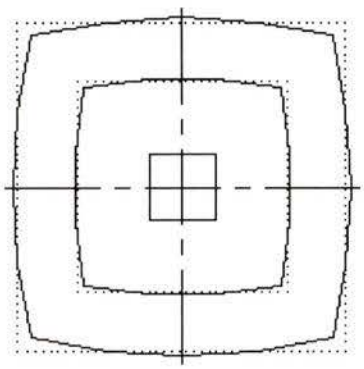


Figure 4.1: Barrel distortion

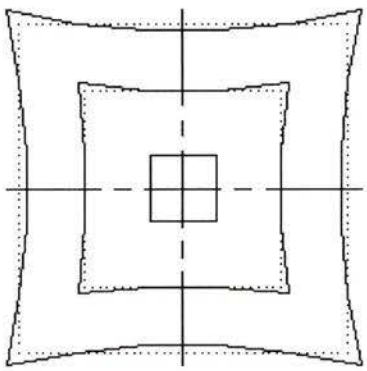


Figure 4.2: Pincushion distortion

Due to distortion, the images do not depict the true shape of the objects under consideration and can lead to unacceptable errors. Perspective distortion will also be present in a system if the camera is mounted at an angle with respect to the object.

4.1.1 Pincushion Distortion

To identify and correct distortions, the camera was mounted with its optical axis perpendicular to the object plane. With this set-up, the images acquired show pincushion distortion. Not only did the camera lens distort the image, but the laser projector added distortion to the laser lines as well. The laser projector was not designed for a working distance of 7-8 feet and hence the laser lines are distorted when projected from such heights. An image taken is shown in Figure 4.3. The pincushion distortion is clearly visible with the corner points spread outwards (like in Figure 4.2).

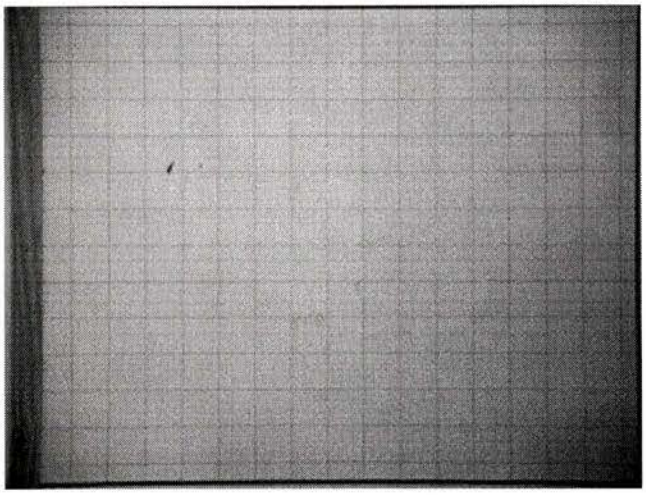


Figure 4.3: Pincushion distortion with camera axis normal to the surface

If the lens distortion problem was solved and then the laser distortion was addressed, much computation time would be wasted in restoring the image. Thus a better approach was taken to solve the problem when both the distortions were present simultaneously.

4.1.2 Methods to Remove Distortion

Various approaches have been devised to remove the distortion in a digital image. Ojanen[14] in his approach, photographed a target sheet and compared the digital image to the actual coordinates. A minimization algorithm was implemented to compute and minimize the deviation.

Fernandes [15] used a look-up table to store the distance of each point in the image from the center most pixel. A multiplying factor was generated to correct the undistorted points and restore the correct coordinates.

A nonlinear optimization approach based on image registration was studied by Tamaki [16]. In their study, parameters like view change, distortion and illumination variation were taken into consideration; however, the computation time was much higher.

Ahmed [17] derived a distortion measure that was optimized using a non-linear search technique. This measure was used to find the best distortion parameters that straighten the lines in a distorted image.

Another approach by Boone [18] aimed to match the undistorted image to the distorted image by using the values from the look-up table. The look-up table was generated by segmenting the image into triangular regions and finding the distance between the markers. The distance between the markers was known and hence provided a benchmark for comparison.

Most of the approaches to solve for distortion are based on optimizing the distance between the distorted and the true points. Such techniques are good if there is no time constraint. Any optimization problem will give best results if the starting point chosen is a good estimate. If the optimization problem does not converge, the distortion parameters computed will be wrong and can produce erroneous results.

For the system under study, distortion in the image is produced by the lens and the laser. Therefore, a new approach which does not require any optimization techniques was developed. Polynomial equations were used to represent the distorted lines which were later restored to the original known positions.

4.1.3 An Approach to Remove Pincushion Distortion

Before starting the experiments, it was ensured that the optical axis of the camera and the laser projector were perpendicular to the table. A square grid on the table was used to align the center lines as viewed from the camera. The square grid had several equally

spaced lines along the x' and y' axis (refer to Appendix II for the coordinate system) with clearly identified center lines. The grid was only used for alignment purposes and not for the distortion measurement.

An image was taken and processed to find if the center lines on the grid correspond to the center of the 640 x 480 pixel computer image. Once it was ensured that the centers lines match, the laser lines were projected on the grid. The projector produced equally spaced parallel lines and the centermost laser line (17^{th} line) was aligned with the center line on the grid. When an image was taken, the distance of each laser line from this center line was fixed. This helped to restore the lines to their true positions.

The following parameters were known for the experiment: measured distance between the lines on the grid were 15 mm and the calculated distance between the lines on the computer image (at the center of the image) was 18 pixels.

With this setup, an image was taken and then the laser projector was rotated by 90 deg and aligned to the center line. Once again and another image was acquired. For the purpose of understanding, the following nomenclature was followed for the projected laser lines.

1. When the laser lines are running parallel to y' axis, we denote them as "vertical lines".
2. When the laser lines are running parallel to x' axis, we denote them as "horizontal lines"

Figure 4.4 below shows the laser lines viewed by the CCD camera in both the directions. It was observed that the distortion was not the same about the centermost line for all the laser lines. In pincushion distortion, one would see approximately equal amount of distortion about the centermost line. This difference can be accounted to the distortion produced by the laser line projector. Therefore, the problem could not be solved by conventional approaches.

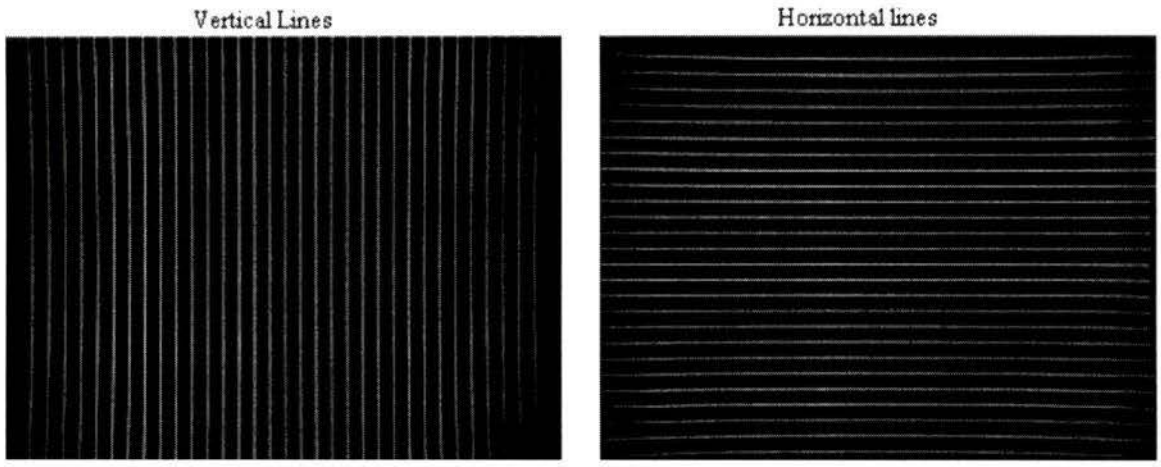


Figure 4.4: Original laser lines showing pincushion distortion in the lens with laser line distortion

The same approach was used to solve the distortion problem for the 'Vertical' as well as the 'Horizontal' lines and therefore the method is explained only for the 'Vertical' lines. The various steps involved in correcting distortion are explained below (for 'Vertical lines')

Step1:

In the image window, the centermost line was taken for comparison because it had no distortion. Since the distance of all the lines from the centermost line was known, lines were plotted on either side at a distance of 18 pixels from the centermost line. If there was no distortion, these lines would overlap the laser lines. However, due to distortion, there was a small difference in the actual and true location of the lines. To represent the distorted line (curve), location of few points lying on that line was found. To get these points, 8 equally spaced lines at y' = 3, 62, 121, 180, 239, 298, 357, 416 and 475 were plotted with the centermost line at the center of the image along with two lines at the ends i.e. y' = 0 and y' = 479. Figure 4.5 below show the distorted lines (curves) and the lines drawn to compute the shift in pixels.

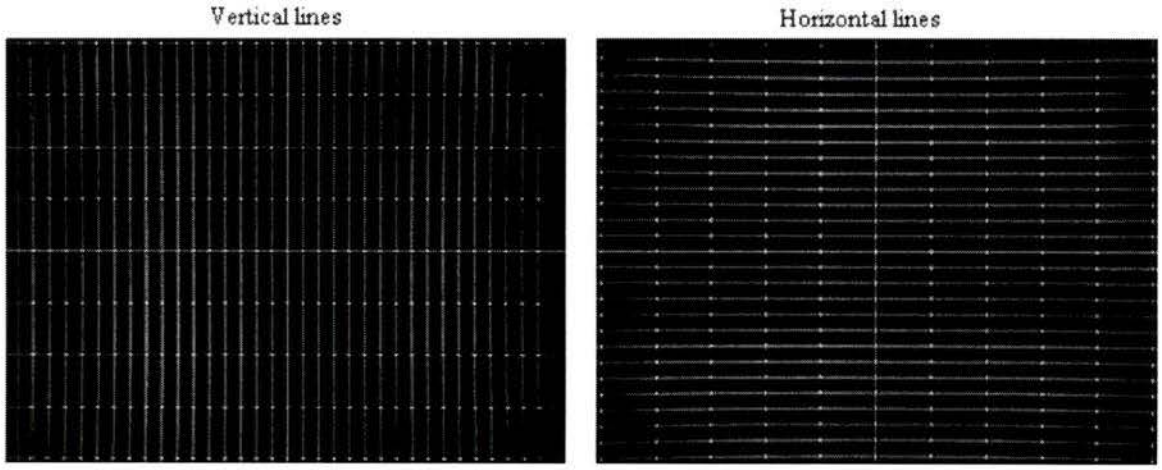


Figure 4.5: Computing pincushion distortion

Step 2

The location of the pixels was found close to the intersection of the lines. For each laser line, the pixel location was taken as a point on the laser line and on the line shown by y' = 0, 3.....479. All the pixel locations were recorded manually and this was repeated over the entire image. Since the laser line is about two-three pixels thick, the centermost pixel was taken as the pixel representing the location.

Though manual data collection can produce errors, it was essential for the current set-up. As mentioned above, the laser line was about two or three pixels wide, and the line width was not consistent at positions where the line (curve) bent. Also due to insufficient brightness of the laser line, it was not possible to segment the lines (to find the centermost pixel) based on the grey scale value, because thresholding led to about 30 – 40% data loss.

The data collected is shown in the table below

Location of the vertical lines in pixels

At y' (pixels)=	Location of the points lying on the distorted line (curve) in pixels																
	31	**49**	**67**	**85**	**103**	**121**	**139**	**157**	**175**	**193**	**211**	**229**	**247**	**265**	**283**	**301**	**319**
	Observed location in pixels																
0	25	44	63	82	101	120	138	156	175	193	211	229	247	265	283	301	319
3	25	45	64	83	101	120	138	156	175	193	211	229	247	265	283	301	319
62	27	46	65	84	102	121	139	157	175	194	212	230	248	266	284	301	319
121	28	48	66	85	103	122	140	158	176	194	212	230	248	266	284	301	319
180	29	48	67	85	104	122	140	158	176	194	212	230	248	266	284	301	319
239	30	48	67	85	104	122	140	158	176	194	212	230	248	266	284	301	319
298	30	48	67	85	104	122	140	158	176	194	212	230	248	266	284	301	319
357	29	48	67	85	103	122	140	158	176	194	212	230	248	266	284	301	319
416	28	47	66	84	103	121	140	158	176	194	212	230	248	266	284	301	319
475	26	45	65	83	102	120	139	157	175	193	212	230	248	266	284	301	319
479	26	44	64	83	102	120	139	157	175	193	212	230	248	266	284	301	319

table continued

At y' (pixels)=	Location of the points lying on the distorted line (curve) in pixels															
	337	**355**	**373**	**391**	**409**	**427**	**445**	**463**	**481**	**499**	**517**	**535**	**553**	**571**	**589**	**607**
	Observed location in pixels															
0	337	355	373	391	410	428	446	464	482	500	519	538	556	575	594	613
3	337	355	373	391	410	428	446	464	482	500	519	537	556	575	594	613
62	337	355	373	391	409	427	445	463	481	499	517	536	554	573	592	611
121	337	355	373	391	408	426	444	462	480	498	517	535	553	571	590	609
180	337	355	373	390	408	426	444	462	480	498	516	534	552	570	589	608
239	337	355	373	390	408	426	444	462	479	498	516	534	552	570	589	608
298	337	355	372	390	408	426	444	462	480	498	516	534	552	570	589	608
357	337	355	373	390	408	426	444	462	480	498	516	535	553	571	590	609
416	337	355	373	391	409	427	445	463	481	499	517	536	554	573	592	610
475	337	355	373	391	409	427	446	463	482	500	519	537	556	574	593	612
479	337	355	373	391	409	427	446	464	482	500	519	537	556	574	593	612

Table 4.1: Points on laser lines – vertical (for pincushion and laser line distortion)

Deviation ranges from six pixels from the corners to none at the center of the image.

Location of the horizontal lines in pixels

Location on the distorted line (pixels)	At x' (pixels) =												
	0	4	67	130	193	256	319	382	445	508	571	634	639
23	19	20	22	23	24	25	25	25	25	24	23	21	20
41	38	38	40	41	42	43	43	43	43	42	41	40	39
59	57	57	59	60	61	61	61	61	61	60	60	58	58
77	75	76	77	78	78	79	79	79	79	78	78	77	76
95	93	94	95	96	97	97	97	97	97	97	96	95	94
113	112	113	113	114	114	115	115	115	115	114	114	113	113
131	130	131	131	132	132	132	133	133	133	132	132	132	131
149	148	149	149	149	150	150	150	150	150	150	150	150	149
167	166	167	167	167	168	168	168	168	168	168	168	168	168
185	185	185	185	185	186	186	186	186	186	186	186	186	186
203	203	203	203	203	203	203	203	203	203	203	203	203	203
221	221	221	221	221	221	221	221	221	221	221	221	221	221
239	239	239	239	239	239	239	239	239	239	239	239	239	239
257	257	257	257	256	256	256	256	256	257	257	257	257	257
275	275	275	274	274	274	274	274	274	274	274	275	276	276
293	293	293	292	292	292	292	292	292	292	292	293	294	294
311	311	311	310	310	310	309	309	309	309	310	311	312	312
329	329	329	328	327	327	327	327	327	327	328	329	330	330
347	347	347	346	345	345	345	345	345	345	346	347	348	348
365	365	365	364	363	363	362	362	363	363	364	365	366	366
383	383	383	382	381	380	380	380	380	381	382	383	384	384
401	402	402	400	399	398	398	398	398	398	399	401	402	402
419	420	420	418	417	416	416	416	416	416	417	419	421	421
437	439	438	436	435	434	434	434	434	434	435	437	439	439
455	458	457	455	453	452	452	451	452	453	454	456	458	458
473	476	476	473	472	470	470	470	470	471	472	474	476	477

Table 4.2: Points on laser lines – horizontal (for pincushion and laser line distortion)

Deviation ranges from four-five pixels from the corners to none at the image center.

Step 3

Once the data is available, different least square polynomial equations were found that represented each distorted line. Since the image was highly distorted at the ends and the values varied by five or six pixels, sixth order polynomial curves were used to represent the distorted lines. However, for most of the lines, third order polynomial curves correctly represented the lines. Starting with the second order polynomial equation, the different data points in each line were checked for compliance. If the equation did not

satisfy all the points, a third order polynomial was created. Sixth order polynomial equations were created to represent some laser lines which had uneven distortion about the center of the image. For these equations, additional points were taken on the curve at known locations, which are not listed in the above table. The table below gives the coefficients of the various curves that were used to represent the lines.

Vertical lines

Loc in pixels	Coefficients for the polynomial equation						
	x^6	x^5	x^4	x^3	x^2	x	Constant
31	0.00	0.00	0.00	-6.18	-8.03	15.14	24.98
49	0.00	0.00	0.00	-3.76	-10.10	13.93	44.50
67	0.00	0.00	0.00	-2.78	-8.74	12.52	63.48
85	0.00	0.00	0.00	3.50	-15.35	12.35	82.52
103	0.00	0.00	0.00	2.65	-13.97	12.40	100.88
121	0.00	0.00	0.00	0.07	-9.08	9.00	120.00
139	0.00	0.00	0.00	2.60	-10.59	9.07	138.01
157	0.00	0.00	0.00	2.60	-10.59	9.07	156.01
175	0.00	0.00	0.00	-3.75	1.13	2.71	174.96
193	0.00	0.00	0.00	0.01	-4.43	4.41	193.10
211	0.00	0.00	0.00	6.35	-11.72	6.36	211.06
229	0.00	0.00	0.00	6.35	-11.72	6.36	229.06
247	0.00	0.00	0.00	6.35	-11.72	6.36	247.06
265	0.00	0.00	0.00	6.35	-11.72	6.36	265.06
283	0.00	0.00	0.00	6.35	-11.72	6.36	283.06
301	0.00	0.00	0.00	0.00	0.00	0.00	301.00
319	0.00	0.00	0.00	0.00	0.00	0.00	319.00
337	0.00	0.00	0.00	0.00	0.00	0.00	337.00
355	0.00	0.00	0.00	0.00	0.00	0.00	355.00
373	-13.23	-56.79	185.09	-160.61	50.32	-4.84	373.02
391	0.00	0.00	0.00	5.30	-4.01	-1.22	391.08
409	0.00	0.00	0.00	-6.40	16.37	-10.95	410.04
427	0.00	0.00	0.00	-6.40	16.37	-10.95	428.04
445	0.00	0.00	0.00	-0.07	9.08	-9.00	446.00
463	0.00	0.00	0.00	-2.90	12.25	-9.83	464.02
481	21.71	-66.06	69.13	-27.15	11.89	-9.50	482.03
499	0.00	0.00	0.00	-0.07	9.08	-9.00	500.00
517	232.31	-744.50	939.74	-578.31	181.46	-30.66	519.08
535	147.79	-478.68	569.03	-303.03	83.94	-19.60	537.58
553	0.00	0.00	0.00	-0.13	16.88	-16.74	556.02
571	-560.00	1636.40	-1813.30	943.00	-206.10	-1.00	575.00
589	-560.00	1636.40	-1813.30	943.00	-206.10	-1.00	594.00
607	0.00	0.00	0.00	-2.71	23.04	-21.40	613.12

Table 4.3: Coefficients for equations – vertical lines (for pincushion and laser line distortion)

Horizontal lines

Loc in pixels	Coefficients for the polynomial equation						
	x^6	x^5	x^4	x^3	x^2	x	Constant
23	0.00	0.00	-34.24	64.35	-57.09	27.93	19.43
41	0.00	0.00	0.00	-1.15	-15.99	18.57	37.99
59	0.00	0.00	-34.42	73.02	-63.83	26.31	56.90
77	0.00	0.00	-17.85	35.78	-34.45	17.51	75.45
95	0.00	0.00	-36.63	73.15	-58.38	22.91	93.36
113	0.00	0.00	0.00	-2.78	-4.88	8.20	112.41
131	0.00	0.00	6.51	-15.34	4.41	5.36	130.49
149	-255.18	697.71	-695.29	305.81	-60.59	8.47	148.46
167	0.00	-30.23	82.95	-77.31	24.22	1.87	166.49
185	0.00	0.00	0.00	-0.04	-2.04	3.15	184.89
203	0.00	0.00	0.00	0.00	0.00	0.00	203.00
221	0.00	0.00	0.00	0.00	0.00	0.00	221.00
239	0.00	0.00	0.00	0.00	0.00	0.00	239.00
257	0.00	0.00	-21.51	34.66	-10.32	-2.95	257.06
275	0.00	0.00	0.00	4.15	0.33	-3.37	274.83
293	0.00	0.00	0.00	4.15	0.33	-3.37	292.83
311	0.00	0.00	0.00	8.18	-2.19	-4.79	310.94
329	0.00	0.00	0.00	-0.12	11.21	-10.00	328.96
347	0.00	0.00	0.00	-0.12	11.21	-10.00	346.96
365	0.00	0.00	0.00	-2.55	16.77	-13.20	365.09
383	0.00	0.00	0.00	-4.23	20.87	-15.66	383.15
401	0.00	0.00	0.00	-2.25	20.96	-18.55	401.97
419	0.00	0.00	0.00	8.13	8.17	-15.09	419.88
437	0.00	-155.89	433.47	-422.54	188.43	-43.04	438.68
455	0.00	-35.35	91.70	-91.37	68.34	-32.88	457.63
473	0.00	0.00	0.00	-2.93	30.74	-27.24	475.97

Table 4.4: Coefficients for equations – horizontal lines (for pincushion and laser line distortion)

Step 4

Once the equations were known, a look up table could be generated to calculate the offset of each of the curves from the original line. However to generate the look up table, more CPU time was required. A different but simple method was implemented to keep CPU time low. For any line, the appropriate equation was used to calculate the location of the point on the curve and restored to the known location of the line. Figure 4.6 shows the corrected image from the distorted images shown in Fig 4.3 & 4.4

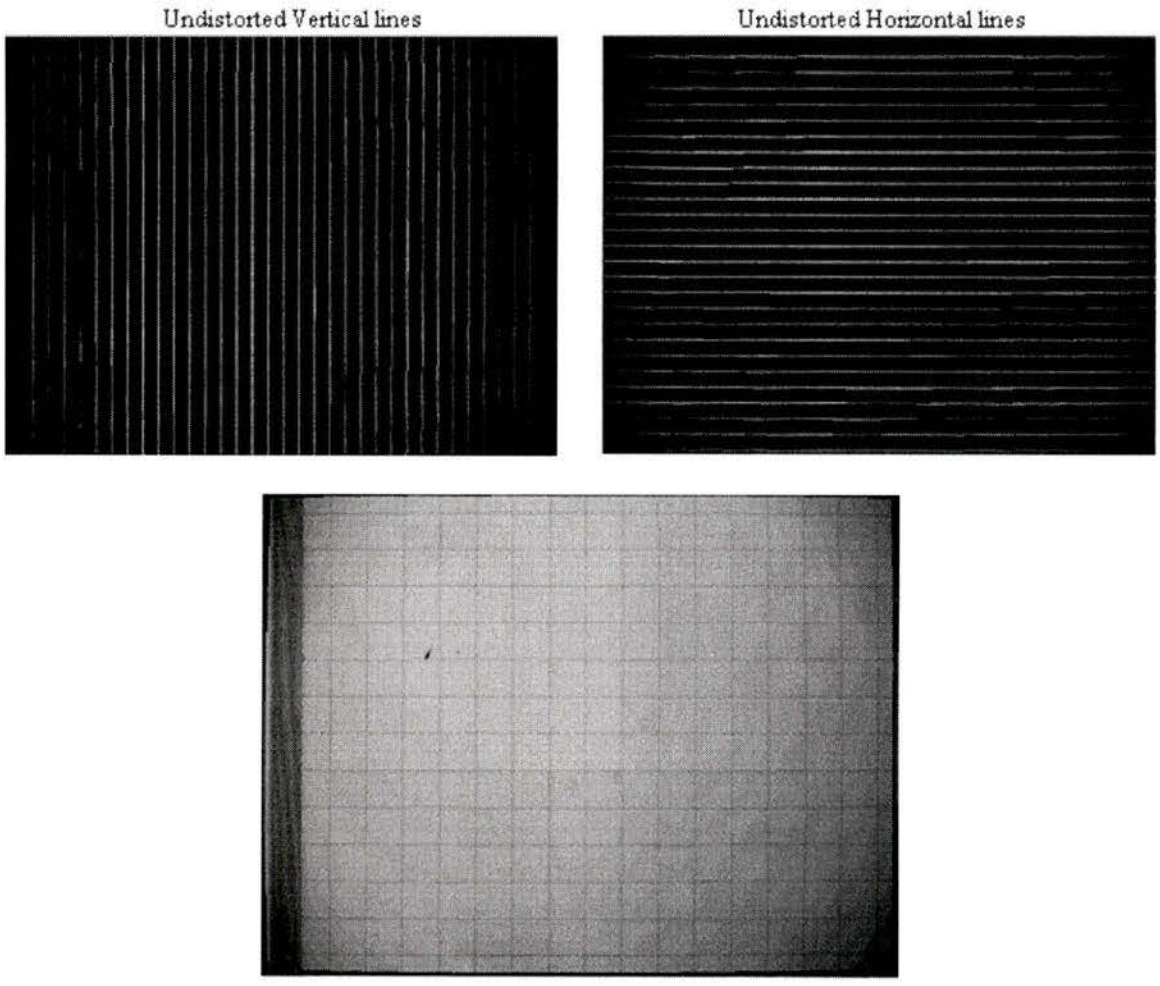


Figure 4.6: Pincushion distortion removed from images

For simplicity and understanding, the first laser line from the 'vertical lines' image is taken for analysis. As seen from the first graph below, the line is represented by a polynomial equation and the pixels on the line have x' values ranging from 25 to 30 in the distorted image. If there was no distortion in the image, the pixels will have an x' value of 31.

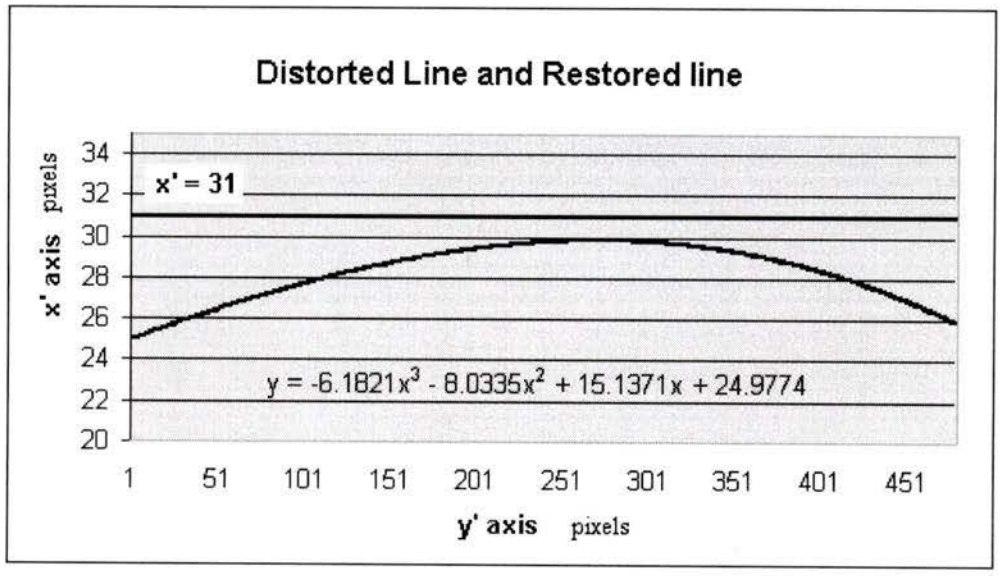


Figure 4.7: Removing distortion

Using the algorithm explained above, the line was restored to the 'true' location which was at an x' value of 31. Since the distortion was consistent towards the ends, the same equation was applied to all the pixels with x' values ranging from 0 to 31. Thus, the image was divided into several regions and for each region, a different equation was implemented.

As seen in the Figure No 4.6, the lines were not distorted anymore, but they looked jagged. This jagged appearance results because the laser line thickness was about two or three pixels. When the data points were recorded manually from the image, a single pixel was chosen which represented the line at that location. However, it was difficult to choose the centermost pixel on a curved line. An attempt was made to take the pixel that was at the center of the curved line, which might induce some error. Using sub-pixel interpolation, this error can be minimized and the weighted grey scale value of the representative pixel can be computed. However, this was not required as the error was corrected when the images were processed for generating the contour. In that process, a data thinning algorithm was implemented that thinned out the line and remove its jagged nature.

4.2 Perspective Distortion Calculation and Compensation

The previous section described the procedure to remove pincushion distortion from the image by mounting the camera in such a way that its optical axis was perpendicular to the table. However, to get the 3D information of any object, the camera is mounted at an angle with respect to the object. The camera angle was fixed to 17° so that all the laser lines were visible from the camera and an image taken is shown in Figure 4.8 below.

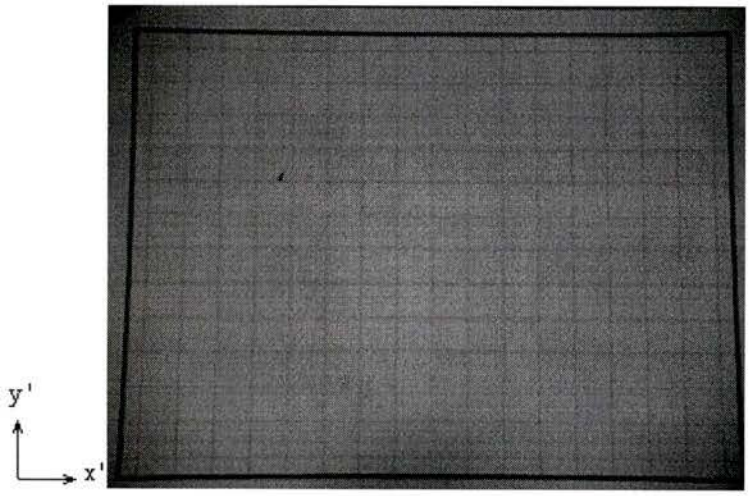


Figure 4.8: Combined pincushion and perspective distortion

One can easily identify pincushion distortion in the image. With the algorithm implemented in the pincushion distortion section, one can see the transformed image in Figure 4.9

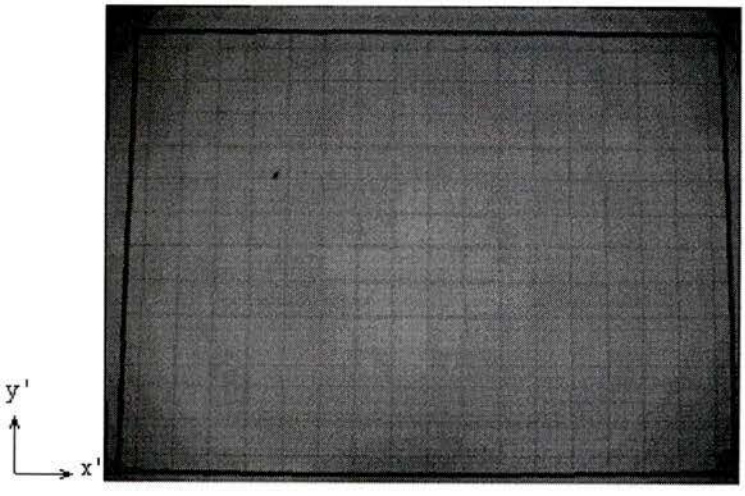


Figure 4.9: Perspective distortion in the image

However, the image did not look perfect because perspective distortion was produced by mounting the camera at an angle. The position of the camera is fixed in all the set-up and hence the perspective error problem was solved in a similar way, as explained in the pincushion distortion section. The laser lines that were projected on the table were skewed due to pincushion and perspective distortion. Figure 4.10 of the laser lines was used to solve for perspective distortion.

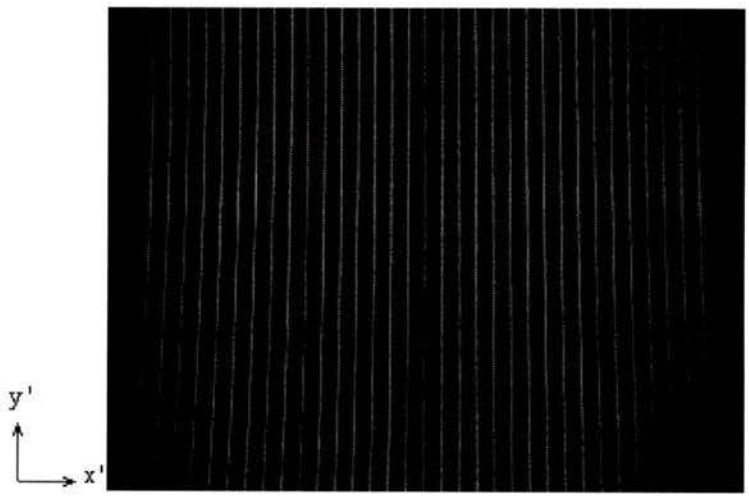


Figure 4.10: Laser lines projected to remove perspective distortion

From the above image, the inter-distance between each of the laser line was known to be 18 pixels and hence the lines that were skewed were restored to their "original" position.

The table below shows the offset of the laser line from the original values.

At y ' (pixels) =	Location of the points lying on the distorted line (Curve) in pixels																
	31	**49**	**67**	**85**	**103**	**121**	**139**	**157**	**175**	**193**	**211**	**229**	**247**	**265**	**283**	**301**	**319**
	Observed location in pixels																
0	49	66	82	100	116	134	151	167	184	201	218	235	252	269	286	301	319
3	49	66	82	100	116	134	151	167	184	201	218	235	252	269	286	301	319
62	48	65	81	99	115	133	151	167	183	200	217	234	251	268	286	301	319
121	47	64	80	98	114	131	148	165	183	200	217	234	251	268	286	301	319
180	44	61	79	95	112	130	147	165	182	200	217	233	251	268	286	301	319
239	42	59	77	94	111	129	146	164	181	199	216	233	250	267	286	301	319
298	40	57	75	92	109	127	145	162	180	198	215	232	249	267	285	301	319
357	37	54	72	90	107	125	143	161	178	196	214	231	249	266	285	301	319
416	34	51	70	87	106	123	141	159	176	194	212	230	247	265	284	301	319
475	32	50	67	85	103	121	139	157	175	193	210	228	246	264	284	301	319
479	32	50	67	85	103	121	139	157	175	193	210	228	246	264	284	301	319

At y' (pixels) =	Location of the points lying on the distorted line (Curve) in pixels															
	337	**355**	**373**	**391**	**409**	**427**	**445**	**463**	**481**	**499**	**517**	**535**	**553**	**571**	**589**	**607**
	Observed location in pixels															
0	336	353	369	386	402	419	436	453	470	487	503	520	537	553	570	587
3	336	353	369	386	402	419	436	453	470	487	503	520	537	553	570	587
62	336	353	370	387	404	421	438	455	472	489	506	522	539	556	573	590
121	336	353	370	387	405	422	439	456	473	490	506	524	541	559	575	592
180	337	354	371	388	405	422	439	457	474	491	508	526	543	561	578	594
239	337	354	371	389	406	423	441	458	476	493	510	528	545	563	580	596
298	337	354	372	390	407	424	441	459	476	494	511	529	547	565	581	599
357	337	354	372	390	407	425	442	460	478	496	513	530	547	566	583	600
416	337	354	372	391	407	425	443	460	478	496	514	531	549	566	584	603
475	337	354	372	391	408	426	443	461	478	497	514	532	550	568	586	605
479	337	354	372	391	408	426	443	461	478	497	514	532	550	568	586	605

Table 4.5: Points on laser lines (for perspective distortion)

There is a huge variation in the actual and the offset value. The pixels were offset by about 20 pixels on the edges. Hence, a polynomial equation was created to represent each line and the equation was used to get the values of the pixels on the laser line which were then restored to the original position. The equation table is shown below.

Loc in pixels	Coefficients for the polynomial equation						
	x^6	x^5	x^4	x^3	x^2	x	Constant
31	0.00	0.00	0.00	12.81	-25.92	-4.04	49.02
49	0.00	0.00	19.22	-19.26	-9.54	-6.57	66.05
67	0.00	0.00	0.00	3.47	-15.16	-3.35	81.93
85	531.30	-1445.30	1414.80	-588.60	83.20	-10.40	100.00
103	0.00	0.00	0.00	-0.35	-4.45	-8.21	116.07
121	0.00	0.00	0.00	-1.90	-1.57	-9.70	134.04
139	0.00	0.00	0.00	-4.62	3.18	-10.89	151.24
157	0.00	0.00	0.00	-1.14	-4.94	-4.09	167.08
175	0.00	0.00	0.00	3.28	-11.11	-1.24	183.87
193	0.00	0.00	62.78	-124.94	68.70	-14.59	201.04
211	0.00	0.00	0.00	-8.64	4.54	-3.95	217.90
229	0.00	0.00	0.00	-9.65	9.38	-6.77	234.98
247	0.00	0.00	0.00	-5.12	3.12	-4.04	251.91
265	0.00	0.00	0.00	-6.10	5.81	-4.74	268.92
283	0.00	0.00	-11.04	36.14	-40.96	15.37	284.44
301	0.00	0.00	0.00	0.00	0.00	0.00	301.00
319	0.00	0.00	0.00	0.00	0.00	0.00	319.00
337	0.00	0.00	0.00	-2.77	2.49	1.29	335.93
355	0.00	0.00	0.00	-2.77	2.49	1.29	352.93
373	0.00	0.00	0.00	-2.74	0.84	4.84	369.04
391	0.00	0.00	0.00	-6.52	8.18	3.31	386.05
409	0.00	0.00	0.00	8.99	-17.97	14.92	402.04
427	0.00	0.00	0.00	6.17	-12.10	12.88	419.09
445	0.00	0.00	0.00	2.39	-7.55	12.20	436.09
463	0.00	0.00	0.00	3.41	-9.61	14.17	453.02
481	0.00	0.00	0.00	-5.73	2.33	11.38	470.07
499	0.00	0.00	0.00	-5.72	5.11	10.52	487.08
517	0.00	0.00	-49.63	90.65	-53.31	23.21	503.02
535	0.00	0.00	0.00	-0.55	-5.94	18.57	519.87
553	0.00	0.00	26.34	-54.17	29.19	11.73	536.98
571	0.00	0.00	32.93	-59.95	21.04	20.99	552.97
589	0.00	0.00	24.57	-42.55	13.67	20.41	569.96
607	0.00	0.00	-15.83	36.19	-27.34	25.10	586.95

Table 4.6: Coefficients for equations – (for perspective distortion)

Once the algorithm was implemented, the perspective error problem was solved and the lines were restored to their original positions. Hence, the corrected image depicted the true shape of the objects under observation without any distortion effects.

For any application, if the position of the camera changes continuously, perspective distortion needs to be corrected for each position. However, in this work, the position of

the camera was fixed and hence the equations created were sufficient to restore the true coordinates.

Figure 4.11 below is the corrected image free of perspective distortion

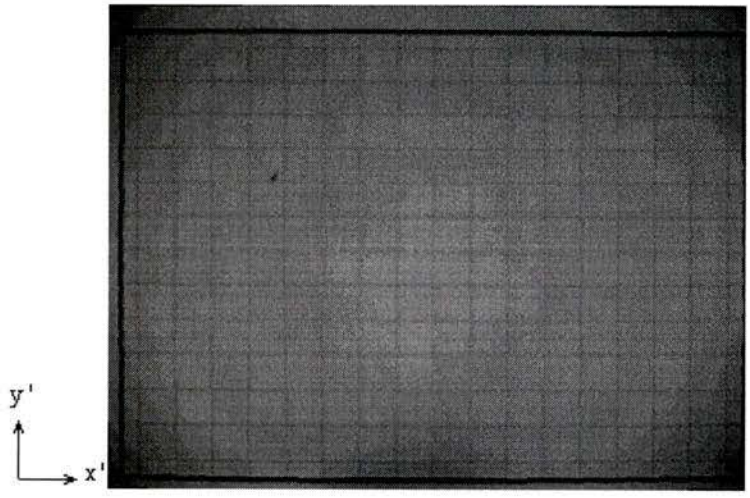


Figure 4.11: Distortion free image

4.3 Calibration Using Known Objects

After the distortion correction algorithms were implemented, the image had no distorted contours. The algorithms relocated the pixels in the image based on the equations derived by the different experiments conducted. The correlation between the size of an object and its actual size in pixels, as measured on the image, needed to be found. The measurement was done on several objects of known dimensions. The scale factors along the x'-axis and the y'-axis were different. The table below shows the calibration data.

Parameter	Object No						
	1	2	3	4	5	6	Avg
True Width in mm	102.83	86.36	183.09	152.51	141.41	75.34	
Width in pixels	112.00	89.00	202.00	168.00	150.00	83.00	
Ratio mm/pixel	0.92	0.97	0.91	0.91	0.94	0.91	0.93
True length in mm	39.82	24.16	29.02	25.12	152.43	102.83	
Length in pixels	48.00	29.00	34.00	30.00	177.00	119.00	
Ratio mm/pixel	0.83	0.83	0.85	0.84	0.86	0.86	0.85

Table 4.7: Calibration parameters

The nomenclature followed for the calibrations is shown in Figure 4.12 below

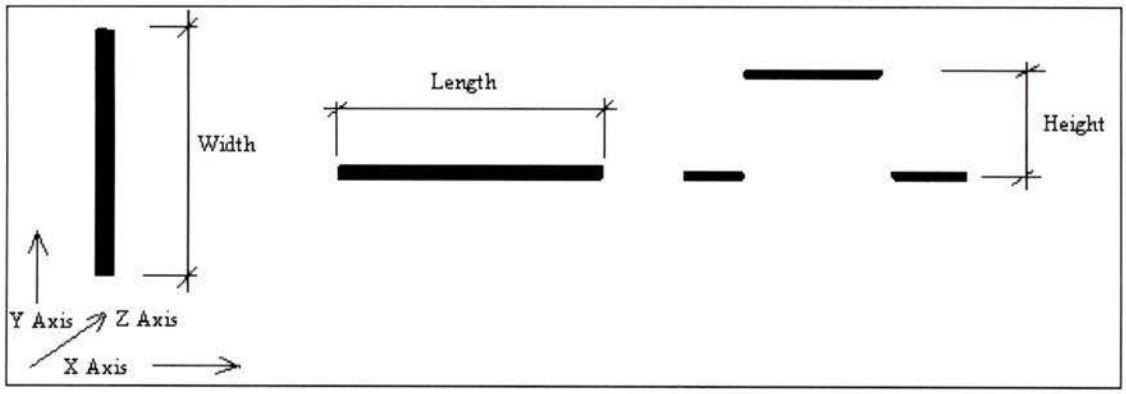


Figure 4.12: Calibration nomenclature

To calculate the height, the distance of the points on the laser line from the camera should be known, i.e. employ triangulation or calibrate each line separately. Out of a total of 33 lines, only 13 lines were used to create the 3D contour of the patient. The reasons are explained in subsequent chapters. Therefore, for each line, calibration was done on standard objects and the following scale factors were calculated.

Line No	Scale factor mm/pixel
8	2.312
9	2.412
10	2.452
11, 14	2.502
12	2.557
13	2.572
15,16	2.522
17	2.552
18	2.592
19, 20	2.603

Table 4.8: Calibration of individual laser line

Chapter 5 Processing the Laser Data

A 2D image captured by the camera was interpreted and processed to generate the 3D contour. This contour was compared to the contour generated from the CT scan which was in 3D point set form. To make this comparison feasible, the camera was mounted at an angle with respect to the bust and an image was captured which has several contour laser lines of the bust. Experiments were done to fix the camera angle and the distance between the camera and the laser. Finally, many algorithms that produced the 3D contour of the bust were implemented.

5.1 Laser Line Processing

The camera was fixed behind the patient (as shown in Appendix II). In the initial experiments, the camera was mounted at 30° with respect to the Y axis. With this set-up, a better 3D contour could be generated and the resolution along the Y axis was much better. But this was not feasible in terms of separating the lines because some of the laser lines were hidden. It was difficult to assume the number of lines hidden and there was no color coding procedure involved to differentiate the lines. Therefore the camera angle was changed to 17° which reduced the resolution along the Y axis but made data processing feasible. The laser projector produced 33 lines and was set in such a way that the 17^{th} line (centermost line) fell on the lead markers. This line was perpendicular to the base. The remaining lines had an inter-beam angle of 0.38° and were therefore inclined.

5.1.1 Thinning the Laser Lines

As mentioned in the previous sections, the laser lines had a thickness of 2-3 pixels. This width was not uniform due to improper lighting and less output power of the laser projector. Also, due to the various algorithms implemented for distortion correction, the laser lines appeared thick and therefore it was essential to find the centermost pixel that represented the line.

To solve this problem, a simple mask was applied to the 2D image which reduced the thickness of the lines. Since the lines run parallel to x' axis, the mask was scanned along the y' axis as shown in Figure 5.1

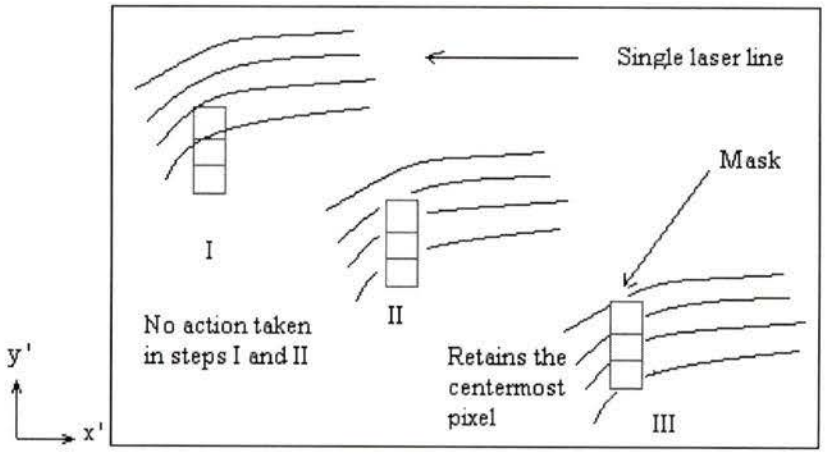


Figure 5.1: Application of the mask for laser line thinning

The mask reduced the thickness of the lines by selecting the centermost point of the line that represents the line. As shown in Figure 5.1, the mask searched for bright pixels on the line along the y' axis of the image. If the previous and the subsequent pixel were bright, the mask retained the centermost pixel and turned the other two black. However if the line thickness was just two pixels, then either of the two pixels were chosen. The corrected image is shown in Figure 5.2.

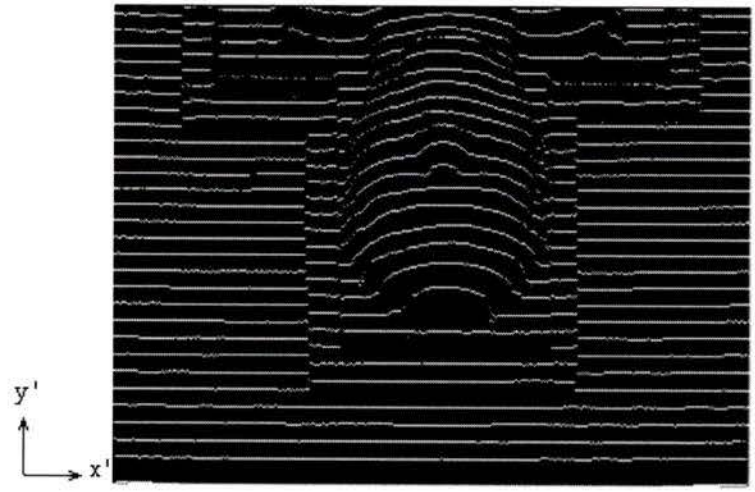


Figure 5.2: Thinned laser lines

5.1.2 Distinguishing the Lines

The laser lines, when viewed from an angle, appear to be merged due to the 3D profile of the object. This made the process of line separation more complex and difficult. Also due to the absence of any reference, the implementation of any algorithm seemed difficult. The laser lines were separated and numbered based on the distance between them. Starting from the origin along y', if a bright pixel was encountered, it was assigned to a line. The laser line was not very bright and had some holes (some pixels on the line were black), therefore some pixels were not detected by the camera. In this section, several algorithms are implemented so that the lines are distinct and only the required part of the data is used rather than processing the entire data.

5.1.2.1 Reference Data Selection & Initial Line Separation

As shown in Figure 5.2, the image was filled with lines but only 20-30 % of the area needed to be processed. This small area helped in generating the contour and the remaining part was discarded. In the software, an option was provided for the user to select the center of the bust, which could be the highest point on each contour line. The user was not expected to select the exact center of the image, but had to approximately locate the center of the bust. With the value defined, about 60 pixels on either side were chosen to give a total of 120 contour points for a single line. The size of every patient varied, but for the current set-up, this width was sufficient to define the shape of the object. Also, if this width was more, the separation process became difficult to implement because the lines were broken or merged beyond this range.

However, reference points from the table surface were required to compute the height of each point on the contour. If the above algorithm was implemented, all the non-contour points were deleted and finding the heights was an impossible task. Prior to implementation of the above algorithm, about 10 pixels from each line that were on the table were stored. As the lines did not merge or overlap on the sides, it was easy to separate the lines and store them in an array.

When the data was selected about the user-selected center point, the lines were separated and the algorithm is given below:

```
for x' = (User selected center - 60) To (User selected center + 60)
        for y' = 479 To 0 Step -1
                if Pixel value > 0 Then
                   store the y' value (selected center) and the x' & y' values separately
                end If
        next
next

'Put the points in an array by using the center-point for comparison
for laser line = 1 To 33
        for point = 1 To 250
                if y' >= user selected value Then
                        store points in an array
                end If
        next point
next line
```

The re-constructed image after implementing the algorithm is shown below in Figure 5.3.

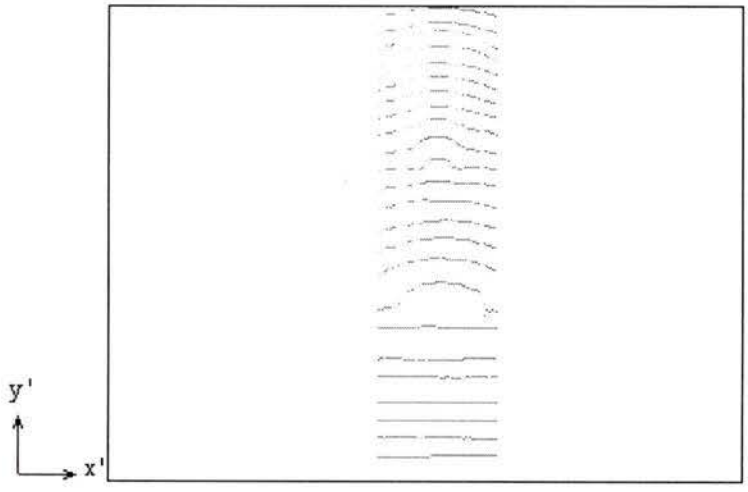


Figure 5.3: Contour after selecting the center-point

Out of a total of 33 lines, only 13 lines were used to represent the bust.

5.1.2.2 Details with Line Separation and Correction

When the algorithm for separating the lines was implemented, the lines were partially separated, and many extra points were associated with these lines. If the lines were continuous without any data gaps, there were no extra points. However, to achieve this, a laser projector of higher power was required. A laser projector of that type could cause permanent damage to the patient's eye and was therefore not recommended. Also,

interpolating the points was not a good option as it led to the creation of a non-accurate contour. The extra points for a particular line are shown in Figure 5.4 below. In this image, the extra points are the points that belong to the adjacent lines.

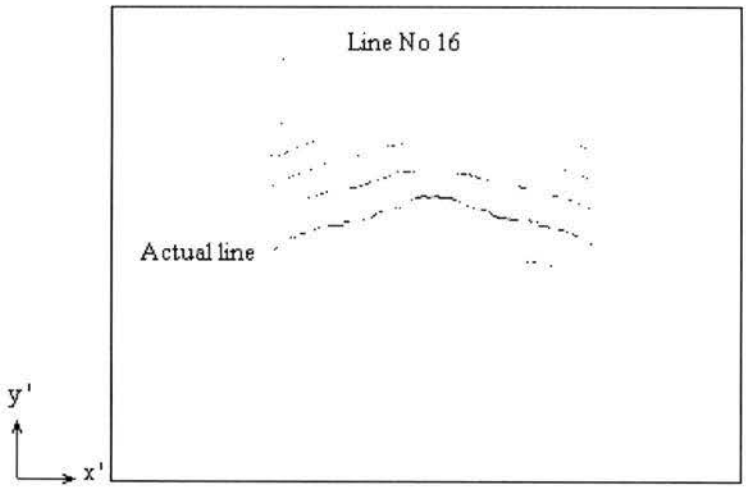


Figure 5.4: A single extracted line

With the center-point selection explained in the previous section, most of these points were deleted as shown in Figure 5.5 below.

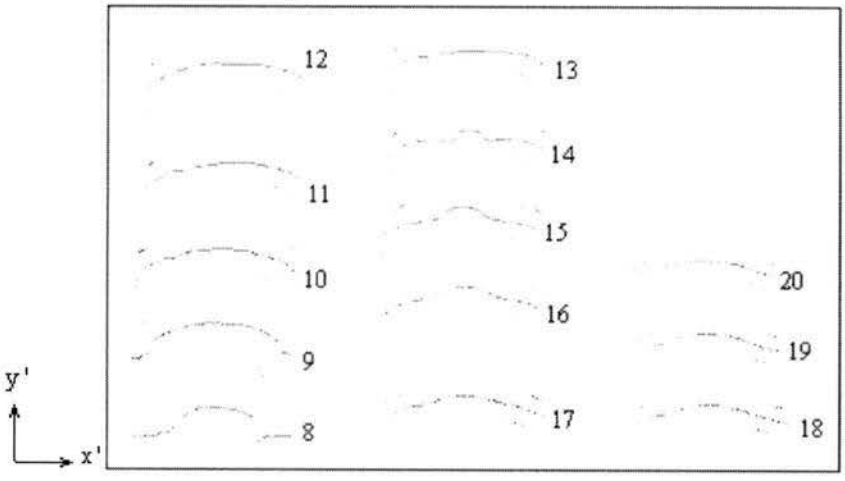


Figure 5.5: Comparing with the centermost pixel

However, a few points remained which were very close to the points of interest (that lie on the contour line) and therefore an grouping algorithm was implemented. The assumptions for the grouping algorithm were,

- The shortest distance between the lines was more than 8 pixels.
- Distance between three lines was always more than 25 pixels

The y' values of all the points were compared with the adjacent points and different groups were created. The group with maximum points was retained and it supposedly formed the group that represented the line accurately. However there was no way to determine if the first pixel selected was on the desired line or the adjacent line. Therefore a total of 10 groups were created and the algorithm was implemented. If fewer groups were taken, the algorithm became sensitive to the initial selection of the bust-center.

Based on the conditions above, the points were separated and different groups were created. Another sorting algorithm was implemented to select the group having maximum points and the points were stored. The improved data is shown in Figure 5.6 below.

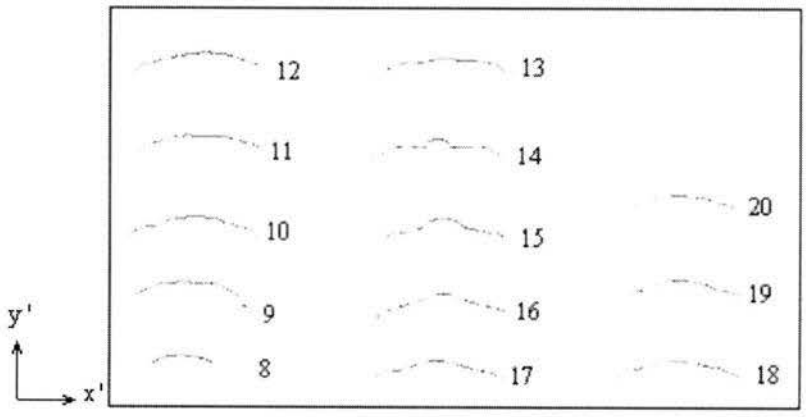


Figure 5.6: Result of implementing the grouping algorithm

5.2 Generating the 3D Contour

In the previous section, the lines were separated and x' and y' values of all the points were stored. Now the system needed to be transformed so that the origin was same as that in the CT set-up. Once this transformation was done, the two 3D data sets were comparable. The three steps in this transformation are explained below.

I: Assigning Z values to the data points.

As discussed in the previous sections, the centermost laser line was aligned to the lead markers. Therefore, Z = 0 was assigned to this line. Since the inter-beam angle was fixed, the lines were, on average, 15 mm apart from each other. Laser line no 1 was closest to the camera and the farthest was no 33. However lines from 8 to 20 were used for the contour generation and the remaining discarded because they did not lie on the bust or a part of the lines was hidden. Based on this concept, 15 mm was added to each line's Z value starting from line no. 17 to 8 and -15 mm was added to each line's Z value starting from line no. 18 to 20.

II: Setting X and Y origin

To find the Y = 0 plane, the lowest Y value on the CT scan contour was found. Taking into consideration the process of CT scan, the lowest Y value corresponded to the top surface of the treatment table. Since the points that lie on the surface in the vision system (section 5.1.2.1) were known, these points were given the lowest CT scan Y value. This ensured that that vision system and the CT data have the same Y origin.

The treatment table moves inside the rotating tube that does the CT scan. All the contour values generated by the CT scan are normalized by a factor in X, Y and Z. The factor for X and Y was 256. The 2D image taken by the camera has 640 pixels along the x' axis. If the centermost 256 pixels are taken then each value in x' was x'-192 (pixels). This defined the X = 0 plane.

III: Applying the calibration factors

The scale factors from the calibration part are incorporated to get the actual X and Y values. These factors are applied to all the data points in the vision system and then normalized to make them comparable to the CT scan data points.

After implementing all the algorithms, the purpose of generating a 3D contour from a 2D image was accomplished. The contour generated is shown below in Figure 5.7 and was used to find the misalignment.

The algorithms used above might generate a different number of points each time and it was based on the selection of the center most point on the image as discussed in section 5.1.2.1.

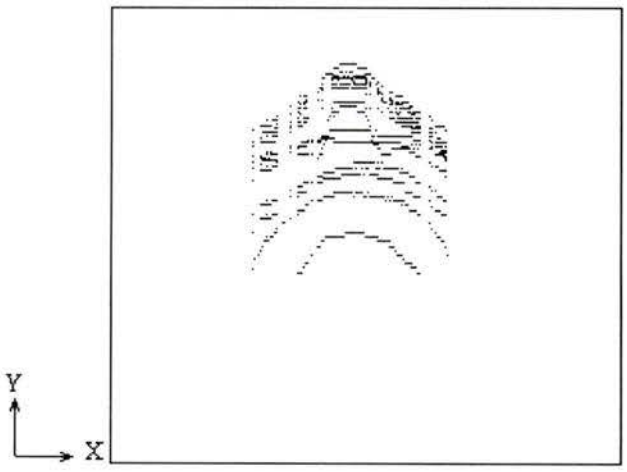


Figure 5.7: Final 3D contour of the bust

The X, Y and Z values of all the points in the vision contour are known. After the 3D vision contour is generated, it is comparable to the CT contour. The CT data and the vision data have the same origin (0,0,0). Since the location of the camera and the laser is fixed, any movement of the bust can be recorded.

Chapter 6 Image Registration, Analysis and Results

The various medical image registration [21] processes aimed at finding and minimizing the difference between two data sets for patient positioning. 3D registration of data sets involves an optimization process, by which the data sets come closer thereby giving the registration parameters i.e. the rotation and translation between the two sets. Registration can be done by either comparing surfaces, triangular facets or points sets. However, it should be noted that the surfaces are created from the points; therefore comparing points will be preferred to surfaces.

Free-form matching using 3D data was done in early days by Faugeras [19] and his group at INRIA, where they used quaternions for an auto part matching. Horn [20] derived a closed form solution to the least-square problem for data point. He formulated a unit quaternion that represents rotation, as the eigenvector associated with the most positive eigenvalue of the symmetric 4 x 4 transformation matrix.

6.1 ICP Algorithm for Image Registration

The Iterative Closest Point (ICP) algorithm was first introduced by Besl [22] in 1992 for image registration. This algorithm aims to minimize the least square distance between a point set to the closest points on the other data sets. There are several advantages of this algorithm over other minimization algorithms and it converges monotonically to the minimum value. Zhang [23] used a k-D trees approach to reduce the computational time in the ICP algorithm to find the closest points. Since the calculation of distance takes tremendous CPU time, the k-D trees was a good approach.

6.1.1 Quaternion Rotation and Computing Translation

The quaternion based algorithms are used in this work, though algorithms based on SVD method [26] are comparable. The basic method of Horn [20], Besl [22] and Faugeras [19] is implemented in this work.

A unit quaternion can be thought of as a combination of a scalar and a vector and is used to represent rotation. Thus,

$$\overset{o}{q} = q_0 + iq_1 + jq_2 + kq_3 \text{ and } q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{6.1}$$

The 3 x 3 rotation matrix generated by a unit quaternion is given by

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix} \tag{6.2}$$

The objective function of the ICP that needs to be minimized is defined as:

$$f(\vec{q}) = \frac{1}{N}\sum_{i=1}^{N} \left\| \vec{CT}_i - R(\vec{q}_R)\vec{Vision}_i - \vec{q}_T \right\|^2 \tag{6.3}$$

where, N is the total number of points from the vision contour; $\vec{CT}_i$ is a vector defining the CT data set; $R(\vec{q}_R)$ is the rotation matrix; $\vec{Vision}_i$ is a vector defining the vision data set; and $\vec{q}_T$ is a translation vector defined by $[q_4, q_5, q_6]$.

The following steps were taken to find the transformation parameters at every iteration:
Step I: Calculate the center of mass of the vision μ_{vision} and CT data μ_{CT}

$$\mu_{vision} = \sum_{i=1}^{N} \vec{Vision}_i \quad \text{and} \quad \mu_{CT} = \sum_{i=1}^{N} C\vec{T}_i \tag{6.4}$$

Step II: Calculate the cross-covariance matrix $\sum_{CT-Vision}$ of both the data sets

$$\sum_{CT-Vision} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\vec{Vision}_i - \mu_{vision}\right)\left(C\vec{T}_i - \mu_{CT}\right)^T \right] \tag{6.5}$$

For more thorough explanation on calculation of the cross-covariance matrix, refer to Horn[20]

Step III: Calculate the symmetric $Q\left(\sum_{CT-Vision}\right)$ 4x4 matrix

$$Q\left(\sum_{CT-Vision}\right)=\begin{bmatrix} tr\left(\sum_{CT-Vision}\right) & \Delta^{T} \\ \Delta & \sum_{CT-Vision}+\sum_{CT-Vision}^{T}-tr\left(\sum_{CT-Vision}\right)I_{3x3} \end{bmatrix} \quad (6.6)$$

where,

Anti symmetric matrix $A_{ij}=\left(\sum_{CT-Vision}-\sum_{CT-Vision}^{T}\right)_{ij}$ is used to find Δ

$$\Delta=\left[A_{23}+A_{31}+A_{12}\right]^{T} \quad (6.7)$$

$tr\left(\sum_{CT-Vision}\right)$ = sum of diagonal elements of the cross-covariance matrix

I_{3x3} = Identity matrix

Step IV: Calculate eigenvalues and eigenvectors of the 4x4 matrix

Referring to the method of Jacobi implemented by Peterson[24], the eigenvalues and eigenvectors of a real symmetric matrix were calculated to give the quaternion. The most positive eigenvalue of the 4 x 4 matrix was used to calculate the desired eigenvector. This eigenvector was the quaternion defined as $\vec{q}_R=\left[q_0 q_1 q_2 q_3\right]$ and it represented the optimal rotation. The translation vector is given by:

$$\vec{q}_T=\mu_{CT}-R(\vec{q}_R)\mu_{Vision} \quad (6.8)$$

Thus the total registration is:

$$q=[\vec{q}_R \mid \vec{q}_T]^t=[q_0 q_1 q_2 q_3 \mid q_4 q_5 q_6]^t \quad (6.9)$$

6.1.2 Implementing ICP

Initially the matrix representing rotation and transformation was initialized.

$\vec{q}_k=[1,0,0,0,0,0,0]^t$

Rotation matrix is $R_k=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$

The 4 steps in the algorithm are explained below.

Step I: Find the closest points.

The number of points in the CT data and the vision data would vary for each implementation. In the software, the user had an option to select the inter-point distance and limit the number of CT points. In the vision system, the selection of the center point determined the number of points available for registration.

To find the closest points, the Euclidian distance between two points i.e. (x_k, y_k, z_k) and $(x_{k-1}, y_{k-1}, z_{k-1})$ is computed. The Euclidian distance is given by:

$$\|dist\| = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 + (z_k - z_{k-1})^2} \tag{6.10}$$

The closest distance between the vision points to the CT points was calculated and stored. The closest distance points were used in the next stage as point set representing the CT data. Though the CT points were more in number, after computing the closest points, the number of CT points was either equal or less than the vision points.

Step II: Compute the registration

The four steps explained in the previous section were implemented to calculate the rotation $[\vec{q}_R]$ and the translation vector $[\vec{q}_T]$. However it must be known that the rotation and the translation parameters would be different at every stage.

Step III: Apply the registration

The total transformation matrix is given by

$$T = \begin{bmatrix} R(1,1)_k & R(1,2)_k & R(1,3)_k & (q_4)_k \\ R(2,1)_k & R(2,2)_k & R(2,3)_k & (q_5)_k \\ R(3,1)_k & R(3,2)_k & R(3,3)_k & (q_6)_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.11}$$

This transformation was applied to the Vision data as follows

$$\begin{bmatrix} Vision_k \\ Vision_k \\ Vision_k \\ 1 \end{bmatrix} = T \begin{bmatrix} Vision_{k-1} \\ Vision_{k-1} \\ Vision_{k-1} \\ 1 \end{bmatrix} \tag{6.12}$$

Step IV: Calculate the objective function

At each stage, the function value was calculated to find the mean square error. If the difference between the function value of the current and previous iteration was below the threshold τ, the algorithm was terminated as shown below.

If $f(\vec{q}_k) - f(\vec{q}_{k-1}) < \tau$ then terminate. (6.13)

After every iteration, the transformation parameters were computed and applied to the vision data set. The mean square distance between the two 3D data sets was minimized at every stage making the algorithm convergent. Thus, the total transformation was a product of all the transformations from every iteration.

In such applications, the difference in the two data sets will be approximately less than 10 mm and/or 10° along any of the axis. Therefore, the time taken for the algorithm to converge depends on the following factors

- Number of CT data points
- Number of vision data points
- Threshold value set to terminate the algorithm
- Accuracy required
- System RAM

The process of computing the closest points needs a lot of CPU time, therefore some modifications were done as explained below:

- Before the ICP algorithm was used to register the data sets, the patient was aligned with the standard techniques. Therefore, it was assumed that the patient

will not be offset by more than 20 mm along the Z axis. If the distance was more than 20 mm, the closest point calculation was not done.

- Similar factors were set along the X and Y axis.

By setting these factors, extra calculations are avoided and that saves a lot of CPU time. Also, since this process was iterative, the aim was to save computation time at each iteration so that the overall time required was less.

6.1.3 Interpreting the ICP Results

The final transformation matrix from the ICP algorithm was interpreted to find the rotation and the translation matrix. Translation can be computed from the last column of the transformation matrix T. Finding the translation vector was easy; however, rotation could be represented in various forms as explained below.

Angle-Axis rotation:
The quaternion can be used to find a unit vector and the rotation angle as shown below.

$$\overset{o}{q} = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\left(iq_x + jq_y + kq_z\right) \tag{6.14}$$

The imaginary part of the unit quaternion is used to calculate the direction of the axis of rotation, and the angle of rotation can be recovered from the real part and the magnitude of the imaginary part. However, this was not useful because the treatment table could not be moved about an arbitrary axis.

X-Y-Z fixed angles:
Another way of defining the orientation of a co-ordinate system is the X-Y-Z fixed angle [25] system. The rotations are performed about a fixed reference frame i.e. a fixed co-ordinate system. The equivalent rotation matrix in the order: $\hat{X}_A$ by γ, $\hat{Y}_A$ by β, $\hat{Z}_A$ by α is given by

$$R_{XYZ} = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma + \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \quad (6.15)$$

This rotation matrix is only true when the rotations are performed in order as explained above. The individual angles can be calculated as:

$$\begin{aligned} \beta &= A\tan 2\left(-R_{31}, \sqrt{R_{11}^2 + R_{21}^2}\right) \\ \alpha &= A\tan 2\left(\frac{R_{21}}{\cos\beta}, \frac{R_{11}}{\cos\beta}\right) \\ \gamma &= A\tan 2\left(\frac{R_{32}}{\cos\beta}, \frac{R_{33}}{\cos\beta}\right) \end{aligned} \quad (6.16)$$

where, $A\tan 2(y,x)$ calculates $\tan^{-1}\left(\frac{y}{x}\right)$ and uses signs of both 'x' and 'y' to determine the quadrant in which the angle lies.

6.2 Analysis & Results

To reduce the computation time, the vision data was reduced based on the user's judgment. When the two data sets were put together, the user could select a cutting plane such that the points lying below this plane were deleted because they did not have any matching points. Various tests were done to check the resolution, accuracy, repeatability and optimization capability of the ICP approach.

6.2.1 Output of the Registration Process

The bust was placed below the laser projector and manually aligned to the center of the frame. An image was taken and the 3D contour of the Vision system was generated and compared with the CT contour. Figure 6.1 corresponds to the two data sets displayed together for registration. The inter-point distance between the CT contour points in this case was 5 mm.

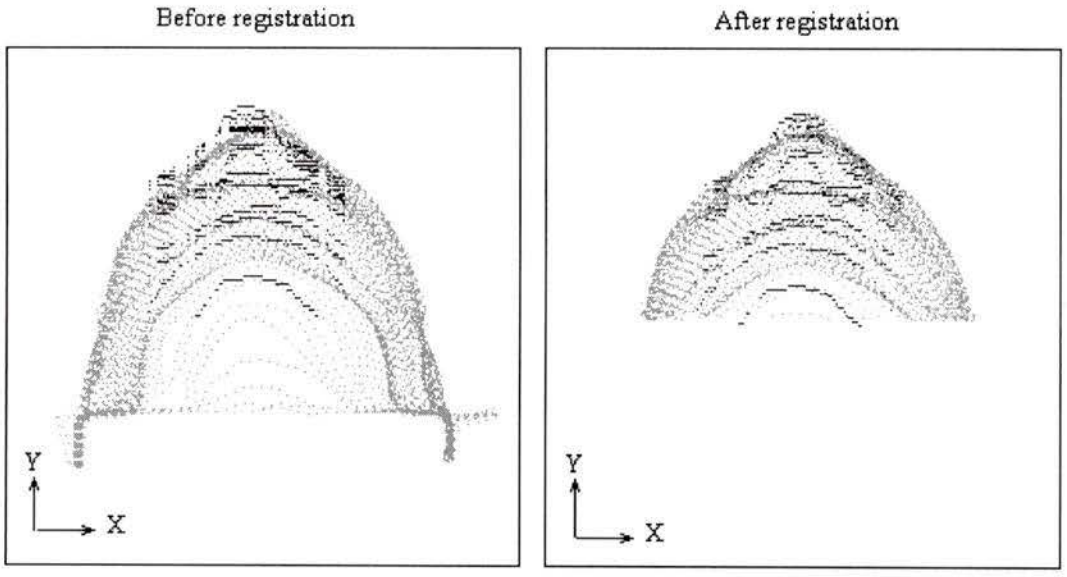


Figure 6.1: Graphical representation of registration

By increasing the inter-point distance, better representation of the contour can be achieved at the cost of CPU time. Different inter-point distances were chosen and the results are summarized in the table below.

Inter-point distance	No. of Vision points	No. of CT points	No. of Iterations	Time taken (min)	Rotation in degrees			Translation in mm		
					X	Y	Z	X	Y	Z
15 mm	1047	1208	17	1 -2	-0.40	-2.54	1.21	10.60	-0.04	-6.71
9 mm	1047	2017	21	2 -3	-0.74	-2.34	1.54	11.32	-1.39	-5.17
5 mm	1047	3644	20	4 -5	-0.78	-2.02	1.47	11.22	-1.26	-4.24

Table 6.1: Results with different inter-point distances

It should be noted that the registration parameters are to be implemented in sequence starting by rotation about X, followed by rotation about Y and Z. The order of translation does not matter and it can be done either before or after rotation.

6.2.2 Global Optimization with ICP

As shown in Figure 6.2, the bust was offset by about 40.00 mm in X along with some small rotation and translations in Y and Z. This will never be the case at the hospital because the preliminary laser aligning methods reduce this to less than 10 mm. However, to check the capability of the algorithm, the bust was shifted to this new position. After a

total of 97 iterations, the algorithm converged and aligned the data sets, giving the rotation as well as the translation parameters in about 9 minutes.

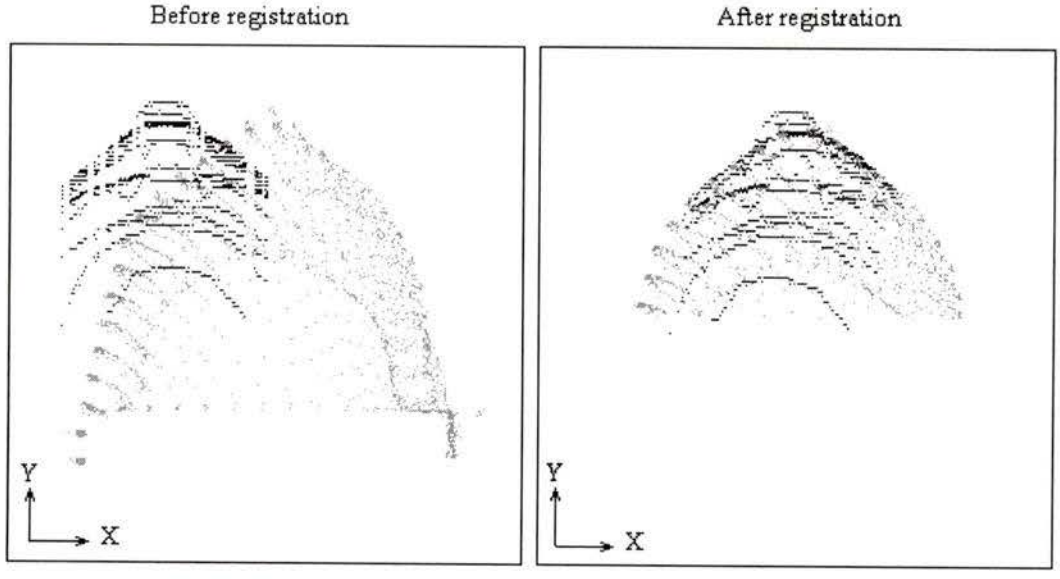


Figure 6.2: Globally optimized approach

6.2.3 Resolution of the Vision System

Resolution is defined as the ability of the system to resolve or detect the smallest distance. In this work, finding the resolution was important because it gave an idea if small movements in the patient's position could be detected. The following parameters should be known before interpreting the images.

- For any patient, over the entire treatment sessions, the treatment table is set to the same height. Therefore the Y values of the data points do not change.
- The laser projector is fixed to the frame and therefore, the Z values for the vision points are assigned to the lines.

For all the resolution measurements, we considered the change in X values. As explained in the calibration section, a pixel value is less than 1 mm and higher than 0.5 mm.

A translation of 1 mm was given to this bust in positive X direction. The following table lists some of the contour values recorded by the algorithm.

Point No	Before translation (pixel)			After translation (pixel)		
	X	Y	Z	X	Y	Z
1	46.75	133.693	135.00	47.60	131.543	135.00
2	111.35	86.715	90.00	112.20	86.715	90.00
3	36.55	60.238	0.00	37.40	60.238	0.00

Table 6.2: Translation along X axis – resolution measurement

A small translation of 1 mm along the X axis can be resolved by the system. A similar test was done to check if a small translation value in Z could be recorded and the results are shown below.

Point No	Before translation along Z (pixel)			After translation along Z (pixel)		
	X	Y	Z	X	Y	Z
1	54.40	129.393	135.00	53.55	129.393	135.00
2	40.80	98.58	105.00	41.65	98.58	105.00
3	27.20	71.545	15.00	26.35	71.545	15.00

Table 6.3: Translation along Z axis – resolution measurement

A movement of ±0.5 mm did not show any changes in the contour values. It can be concluded from the above tests that a small displacement along the X and Z axis, produce changes in the contour values and so the algorithms are sensitive to such changes. The resolution of the vision system is 1.00 mm along the X and Z axis. As explained above, the translation about Y is not a matter of concern because the test bed is set to the same position each time.

6.2.4 Accuracy of the Vision System

Accuracy of any system is its ability to measure the dimension of any object correctly. To check the accuracy of the vision system, a standard object was placed in the field of view of the camera and its length was measured. The accuracy check was not done along the Z axis because the values are not recorded or measured but assigned to the individual laser lines.

Sr. No	Actual dimension in mm	Locations of the object edges – pixels		Measured dimension in Pixels	Measured dimension in mm	Difference in mm
		First edge	Second edge			
1	106.14	198	323	125	106.25	0.11
2	106.14	369	493	124	105.4	0.74
3	106.14	614	743	125	106.25	0.11
4	106.14	451	577	126	107.1	-0.96

Table 6.4: Accuracy measurement along X axis

The measured values are close to the actual values and the range is from -0.96 to + 0.74 mm. Therefore, it is concluded that the vision system is accurate within ±1.00 mm along the X axis. To find the accuracy along the Y axis, a height of a rectangular box was measured at different locations.

Sr. No	Actual dimension in mm	Locations of the object edges in pixels		Measured dimension in Pixels	Measured dimension in mm	Difference in mm
		First edge	Second edge			
1	133.71	230	283	53	132.60	1.11
2	133.71	199	250	51	131.17	2.54
3	133.71	100	151	50	132.70	1.01

Table 6.5: Accuracy measurement along the Y axis

The accuracy along the Y axis is not good because the camera is mounted 8 feet above the table. Therefore, along the Y axis, the accuracy of the system lied in the range of 0 to +2.54 mm.

6.2.5 Repeatability

Any system is considered reliable if it is consistent and produces the same results under the same conditions. The algorithms were checked for repeatability by the following test. The bust was moved by +15 mm in X direction and the contour values were recorded for the set-up. The bust was then moved back to the original position and the contour values were recorded.

Point No	Original Position in pixels			Translation in X by +15mm (in pixels)			Restore to original position (translate in X by -15 mm) - (in pixels)		
	X	Y	Z	X	Y	Z	X	Y	Z
1	59.50	99.29	135.0	72.25	99.29	135.0	59.50	99.29	135.0
2	62.05	70.42	90.0	75.65	70.42	90.0	62.90	70.42	90.0
3	111.35	60.24	0.00	124.1	60.24	0.00	112.20	60.24	0.00

Table 6.6: Repeatability check – movement along the X axis

A similar test was done in Z direction and the bust was removed from its original position and then placed again in the same position. The results of this test are shown in the table below:

Point No	Original Position in pixels			Restore to original position in pixels		
	X	Y	Z	X	Y	Z
1	99.45	97.14	135.0	98.60	97.14	135.0
2	152.1	74.18	75.0	151.3	74.18	75.0
3	112.2	44.93	-30.0	112.2	44.93	-30.0

Table 6.7: Repeatability check – movement along the Z axis

In both the tests, the bust was moved and then the previous position was restored. On checking the contour values, it was found that there was very small variation in the X and Y values, in the amount of ±1.00 mm maximum. The small difference can be accounted to the errors in manually moving the bust.

6.2.6 Testing on Other Phantom

The different tests carried out on the bust gave good results and it was decided to check the performance of the algorithms on other phantom. For experimental studies a phantom resembling a breast cancer patient was considered. CT scan of the phantom was done to get the diagnostic data. The phantom was then placed under the vision system to compute the misalignment. To check the effectiveness of the approach, registration algorithm was implemented to calculate the offset. Figure 6.3 below shows the offset and a good registration after the ICP algorithm was implemented.

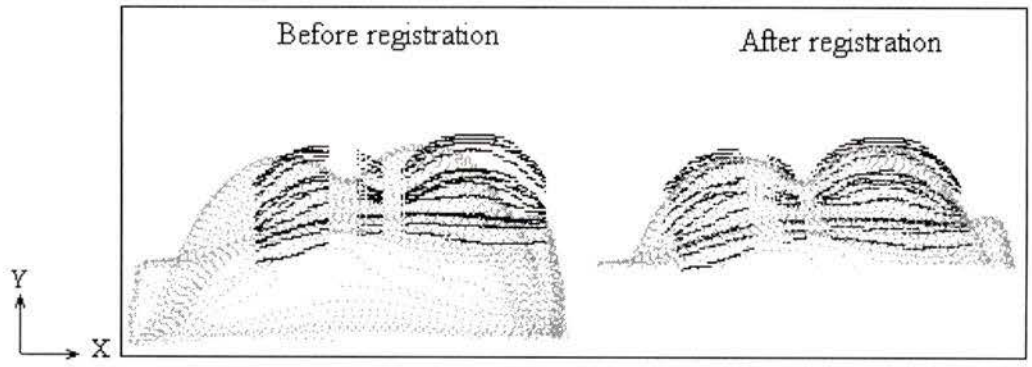


Figure 6.3: Phantom registration – a graphical representation

The output of the registration process was:

Rotation in degrees			Translation in mm			Total Iterations	Time taken
X	Y	Z	X	Y	Z		
-0.77	3.31	-0.56	-21.77	0.758	7.88	29 nos.	9 min.

Table 6.8: Registration output for the phantom

It can be seen that that this algorithm can be used not only for positioning the head –neck cancer patients but also the breast cancer patients.

Chapter 7 Conclusion & Future Work

This work aims to demonstrate the above mentioned registration technique. This technique has many advantages over the standard positioning techniques. Perhaps, the most important is avoiding the use of radiation for patient positioning. The algorithms have been tested in worst positioning cases of the bust and the phantom. The registration is good. Various steps have been taken to reduce the computation time, but due to the inherent nature of the Visual Basic package, the average time taken for the entire registration process was around 10 minutes. With the firewall and antivirus software installed, a slower processor, the system became slower and this added to the increased computation time. The average time taken for any other registration technique varies according to the skill and availability of equipment for positioning but, on average the radiologist spends almost half an hour in this process. With the vision approach, the time taken can be considerably reduced and the patient can be easily and accurately positioned in about 10 minutes with +/- 1.00 mm accuracy along the X axis and 0 to +2.54 mm along the Y axis. Reducing the camera-object distance or incorporating another camera can reduce this error.

Tests were done in the Optics Lab at the University of Victoria to determine the effectiveness of the methods in terms of accuracy, resolution and repeatability. Testing of positioning on people would be a good next step in the hospital environment.

The transformation parameters are defined about the origin (X-Y-Z); however, the movement of the treatment table is feasible about the iso-center of the table. The coordinates of the iso-center are recorded in the CT scan and therefore, the frame can be transformed to give the new parameters to find the final transformation matrix.

Some of the factors that can improve the process are given below.

- Using a lens which has less or no distortion – telecentric lenses.
- Using a 15 line laser projector with higher inter-beam angle, so that line separation is not a problem and designing the laser projector so that the lines are not distorted and higher level harmonics are removed.

Appendix I Specifications of the Equipment Used

Camera - NEC T1 324A

Type	:	Interline transfer CCD
Picture elements	:	768 (H) x 493 (V)
Sensing area	:	8.8mm x 6.6 mm

Varifocal Lens –Tokina TV lens

Max CCD Format	:	2/3"
Aperture (f/#), C = closed	:	F1.8 –22 C
Focal length	:	12.5 – 75.0 mm
Zoom	:	Manual

Frame Grabber

Make	:	Coreco Imaging Bandit II CV
Video controls	:	Brightness, contrast, hue, & saturation adjustments
Formats supported	:	NTSC, RS170, PAL & CCIR
Cable	:	OC-B20C-4RGB0

Video Capture Software – Sapera LT 4.1
Sapera Active X controls for capturing live video or save images in Visual Basic

Multiple-line laser

Model	:	SNF-533L-670-10-15deg-0.38
Number of lines	:	33 Nos.
Fan angle	:	15°
Inter-beam angle	:	0.38°
Wavelength	:	673.90 nm
Laser power	:	7mW
Current	:	67 mA
Class CRDH	:	IIIa

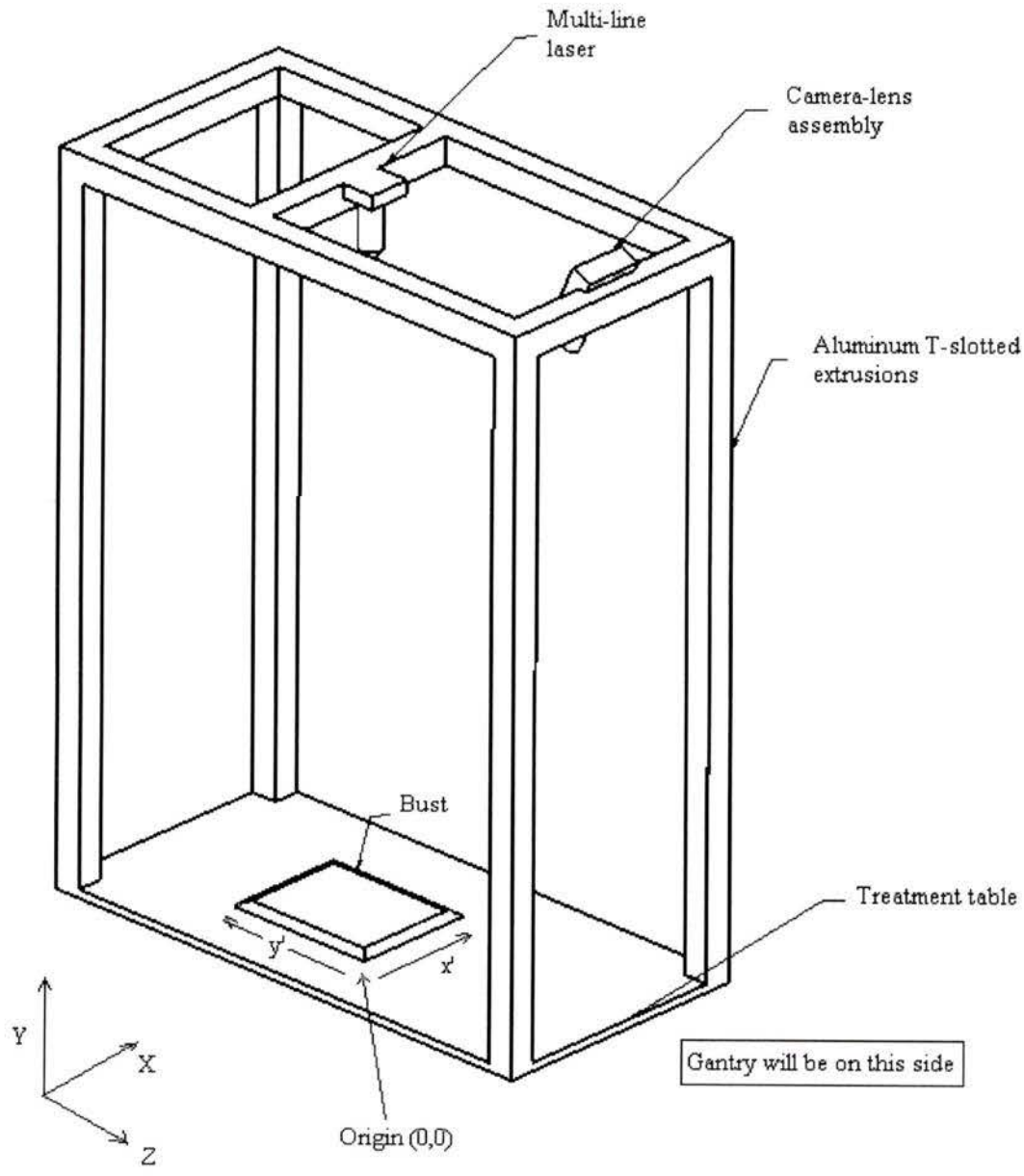
Computer

RAM	:	96 MB RAM
Processor	:	Pentium III at 650 MHz
Operating System	:	Windows 2000

Monitor

Screen Area	:	1024 x 768 pixels

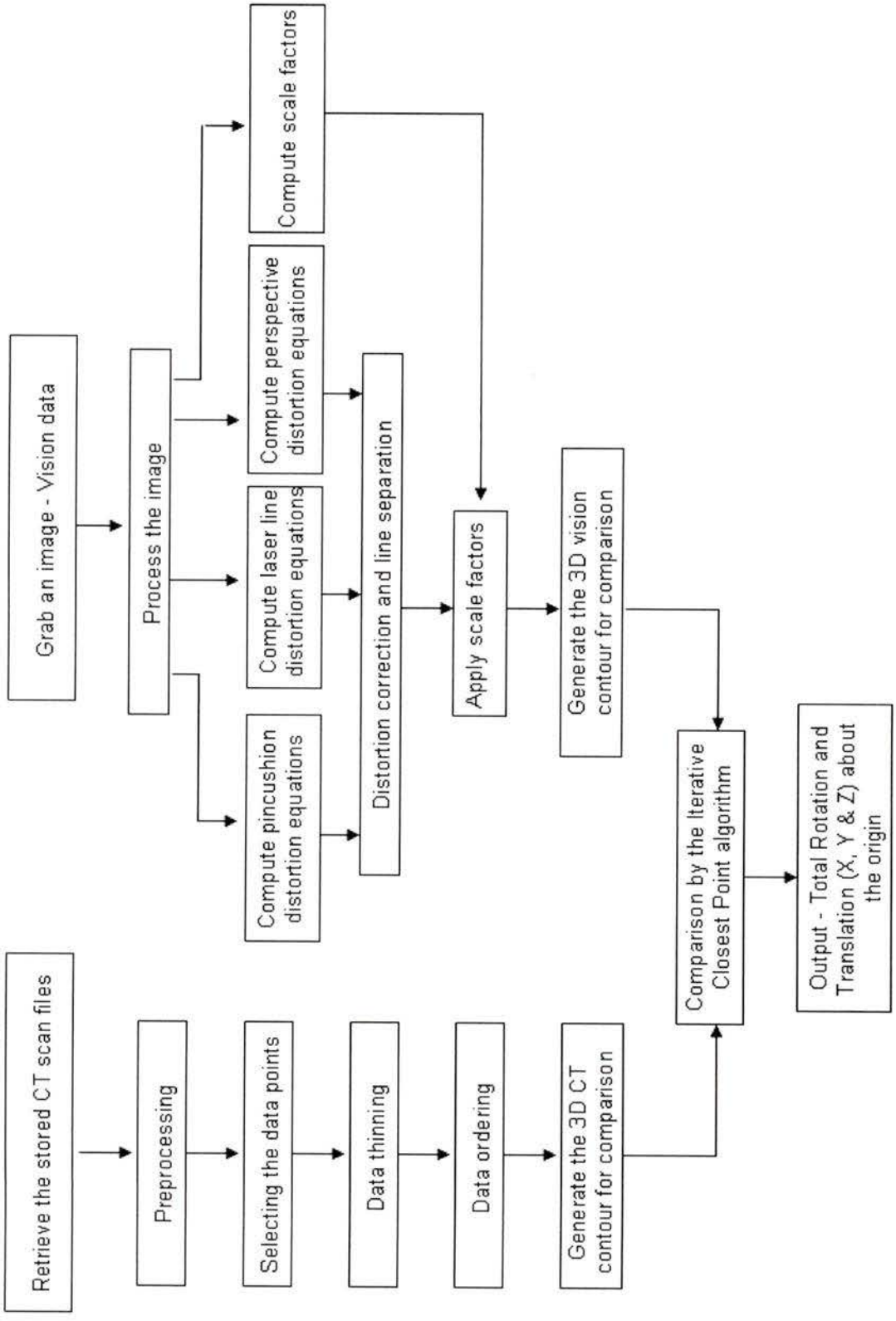
Appendix II Vision Set-up



X, Y, Z is the co-ordinate system of the CT Scan

x'-y' are the co-ordinates of the 2D picture taken by the camera

Appendix III Flowchart



References

1. Baroni, G., Ferrigno, G., Orecchia, R., and Pedotti, A., "Real-time three-dimensional motion analysis for patient positioning verification," *Radiotherapy and Oncology*, 54 (2000) pp. 21-27.

2. Lerch, I.A., and Barish, R.J., "Development of optical process for accessing three-dimensional patient topography," *Medical Physics*, Vol. 5, No. 6, Nov/Dec 1978, pp. 546-549.

3. Johnson, L.S., Milliken, B.D., Hadley, S.W., Pelizzari, C.A., Haraf, D.J., and Chen, G.T.Y., "Initial clinical experience with a video based patient positioning system," *International Journal of Radiation Oncology, Biology, Physics*, Vol. 45, No. 1, pp. 205-213, 1999.

4. Milliken, B.D., Rubin, S.J., Hamilton, R.J., Johnson, L.S., and Chen, G.T.Y., "Performance of a video-image-subtraction based patient positioning system," *International Journal of Radiation Oncology Biology, Physics*, Vol. 38, No. 4, pp. 855-866, 1997.

5. Van Lin, E.N.J., Nijenhuis, E., Huizenga, H., Van Der Vight, L., and Visser, A., "Effectiveness of couch height-based patient set-up and an off-line correction protocol in prostrate cancer radiotherapy," *International Journal of Radiation Oncology, Biology, Physics*, Vol. 50, No. 2, pp 569-77, 2001.

6. Schewe, J., Lam, K., Balter, J., and Haken, R.T., "A room-based diagnostic imaging system for measurement of patient set-up," *Medical Physics*, Vol. 25, No. 12, December 1998.

7. Graham, P.A., Moore, C.J., and Mackay, R.I., "Dynamic surface matching for patient positioning in radiotherapy," Proc Int Conf Information Visualization 1998, pp16-25.

8. Kreb, J., Minohara, S., Endo, M., Debus, J., and Kanai, T., "Patient positioning verification using CT images," *Medical Physics*, Vol. 26, No. 6, June 1999.

9. Cai, J., James, C.H., Saxena, A., and Lanzi, L.H., "A simple algorithm for planar image registration in radiation therapy," *Medical Physics*, Vol. 25, No. 6, June 1998.

10. Thirion, J.P., Gourdon, A., Monga, O., Guezeic, A., and Ayanche, N., "Fully Automatic registration of 3D Cat Scan images using crest lines," Engineering in Medicine and Biology Society, Vol.14. Proceedings of the Annual International Conference of the IEEE, Vol. 5, Oct/Nov 1992, pp. 1888 -1890.

11. Berry, J.A., and Aldrich, J.E., "Surface Topography for patient repositioning," *Medical Dosimetry*, Vol.16, pp 71-77, 1991.

12. Bradley, C., "Rapid prototyping models generated from machine vision data," *Computers in Industry* 44 (2001) pp. 159-173.

13. Lee, K.H., Woo, H., and Suk, T., "Point Data reduction using 3D Grids," *International Journal of Advanced Manufacturing Technology* (2001) Vol. 18, pp. 201-210.

14. Ojanen, H., "Automatic correction of lens distortion by using digital image processing," http://www.iki.fi/hojanen/undistort/

15. Fernandes, J.C.A., Ferreira, M.J.O., Neves, A.B.C. and Couto, C.A.C., "Fast correction of lens distortion for image applications," Proceedings of the IEEE International Symposium on Industrial Electronics, Vol. 2 , Jul 1997, pp 708-712.

16. Tamaki, T., Yamamura, T., and Ohnishi, N., "Correcting distortion of image by image registration," The 5th Asian Conference on Computer Vision, pp. 23-25 Jan 2002, Melbourne, Australia.

17. Ahmed, M., Farag, A., "Non-metric Calibration of Camera Lens distortion," Proceedings of International Conference on Image Processing, Vol. 2, Oct 2001.

18. Boone, J., Seibert, J.A., and Barret, W.A., "Analysis and correction of imperfections in the image intensifier-IV-digitizer imaging chain," *Medical Physics*, 18(2), Mar/Apr 1991.

19. Faugeras, O.D., and Herbert, M., "The representation, recognition, and locating of 3-D objects," *International Journal of Robotic Research*, Vol. 5, No. 3, pp. 27-52, Fall 1986.

20. Horn, B.K.P., "Closed-form solution of absolute orientation using unit quaternions," *Journal of Optical Society America A*, Vol. 4, No. 4, pp. 629-642, April 1987.

21. Hill, D.L.G., Batchelor, P.G., Holden, M., and Hawkes, D.J., "Medical Image registration," *Physics in Medicine and Biology* 46 (2001) R1-R45.

22. Besl, P.J., and Mckay, N.D., "A method for registration of 3-D shapes," *IEEE transaction on pattern analysis and machine intelligence*, Vol. 14, No. 2, Feb 1992.

23. Zhang, Z., "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, 13:2, pp. 119-152, 1994.

24. Peterson, L.E., "Partitioning large-sample microarray-based gene expression profiles using principal components analysis," *Computer Methods and Programs in Biomedicine* 70(2): 107-119, 2003.

25. Craig, J., "*Introduction to Robotics: mechanics and control*," Addison-Wesley Publishing Company Inc., 1986.

26. Arun, K.S., Huang, T.S., and Blostein, S.D., "Least-Squares fitting of two 3-D point sets," *IEEE transactions on pattern analysis and machine intelligence*, Vol. PAMI-9, No. 5, Sept 1987.

Vita

Surname: Shukla Given Names: Salil

Place of Birth: Nasik, Maharashtra, India

Educational Institutions Attended:

University of Victoria	2001 – 2003
Maharashtra Institute of Technology	1994 – 1997
D Y Patil College of Engineering	1993 – 1994

Degrees Awarded

B.E., Mechanical Engineering	1997

Honours and Awards:

The Mr. Alfred Smith and Mrs. Mary Anderson Scholarship	2002 – 2003
The Mr. Alfred Smith and Mrs. Mary Anderson Scholarship	2001 - 2002
University of Victoria Fellowship	2001 – 2002

University of Victoria Partial Copyright License

Title of Thesis:

A Vision System for Patient Positioning in Radiation Therapy

Author: [redacted]

Salil Shukla

June 18, 2003