3-D Coordinate Estimation Model

K.A. Arthur MSc PhD CPhys MInstP
Brandt Technologies Ltd.
<kieran@brandttechnologies.com>

Date: Fri 30/06/2006

Abstract

The purpose of this report is to present a description of the research conducted into camera calibration and the extraction of 3-D coordinate data from images.

Contents

List of Figures

Chapter 1

Literature Review

1.1 Introduction

The purpose of this section is to outline research in the area of camera calibration and data extraction from images carried out by Brandt Technologies Ltd. Brandt Technologies Ltd. were commissioned by the University of Ulster as an industrial research collaborator through the MIDAS initiative. The content relates to a larger body of digital human research by Justin Magee (Art and Design Research Institute) from the School of Creative Arts, Magee Campus, Derry. Brandt Technologies and Justin Magee continue with ongoing research collaborations.

1.2 Definitions

"Photogrammetry is the art, science, and technology of obtaining reliable information about physical objects and the environment through the processes of recording, measuring, and interpreting photographic images and patterns of electromagnetic radiant energy and other phenomena." See Ref [37].

"Photogrammetry can also be thought of as the sciences of geometry, mathematics and physics that use the image of a 3D scene on a 2D piece of paper to reconstruct a reliable and accurate model of the original 3D scene." See Ref [37].

Photogrammetric calibration: Camera calibration is performed by observing a calibration object whose geometry in 3-D space is known with high precision. The calibration object usually consists of two or three planes orthogonal to each other. Sometimes, a plane undergoing a precisely known translation is also used. These approaches require an expensive calibration apparatus, and an elaborate setup. Small errors in calibration apparatus can cause significant errors in the output.

Self-calibration Ref [40]: Techniques in this category do not use any calibration object in the scene. By moving a camera in a static scene, the rigidity of the scene provides in general two constraints on the cameras internal parameters from one camera displacement by using image information alone. Therefore, if images are taken by the same camera with fixed internal parameters, correspondences between three images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3-D structure up to a similarity. Because there are many parameters to estimate, we cannot always obtain reliable results.

1.3 Process overview

An overview of the process of reconstruction is given in a number of references, such as Ref [35]:

1. Feature detection, that is detection of pixels points whose 3-D coordinates are known.

2. Camera calibration, using the above reference calibration data.

3. Feature tracking (Image registration), using information about the same 3-D features in two or more images.

4. Scene reconstruction, given other points in the image, identify the 3-D coordinates.

This list of actions is fundamental and we will not be differing from it substantially. From Ref [36] we note that there is a distinction between feature detection and feature extraction. The authors' first definition is that "Feature detection aims at finding the position of features in the images". Their second definition is that for "feature extraction, a position in the images is described by a set of features, which characterize the local neighborhood". We shall be implementing both types of activities in the software. In the following sections we shall see what the literature says about implementing 3-D coordinate data extraction from images.

1.4 Literature review

1.4.1 Camera Calibration

The purpose of this section is to outline the content of a selection of papers related to camera calibration and scene reconstruction from stereo images. It is not the purpose of this report to provide an overview of this whole area of research and the current state of the art.

A standard reference work on numerical programming, Ref [21] was used for the implementation of a minimization routine. It has some critics, see Ref [27], but it is a good starting point for many problems. Ref [34] is the companion book to Ref [21], and is essential for setting up the programs in the foregoing.

One of the most frequently cited papers in the area of camera calibration is by Tsai; Ref [29]. The author offers both a critical examination of the current state of the art and a method by which 3D robot vision might be improved.

The author, describes the problem of camera calibration for the purposes of 3D vision as, "the process of determining the internal camera geometric and optical characteristics *(intrinsic parameters)* and/or the 3D position and orientation of the camera frame relative to a certain world coordinate system *(extrinsic parameters)*". The author takes a general view of the use of this data, and, for the purposes of this study we are only interested in "inferring 3D information from computer image coordinates". In order to be useful in an industrial environment, certain criteria for the calibration process are set:

1. Autonomous - not requiring operator intervention.

2. Accurate - theoretical process should model experimental setup to the fullest extent, that is, to include the nonlinear nature of lens distortion.

3. Efficient - within reason should be high speed.

4. Versatile - should operate uniformly for different applications.

The paper describes the camera as a pinhole camera model of perspective projection. Given the position of a point in 3-D space, the model predicts the location of the point in the image in 2-D coordinates. The model has 11 parameters. Later in this report we will use software which implements the Tsai algorithm, Ref [38]. The list of parameters is taken from Ref [38], and the notation is taken from Ref [29].

1. f - effective focal length of the pin hole camera.

2. κ_1 - 1st order radial lens distortion coefficient.

3. C_x, C_y - coordinates of center of radial lens distortion -and- the piercing point of the camera coordinate frame's Z axis with the camera's sensor plane.

4. s_x - scale factor to account for any uncertainty due to framegrabber horizontal scanline resampling. This is discretization error due to pixelisation by the sensor.

5. R_x, R_y, R_z - rotation angles for the transform between the world and camera coordinate frames.

6. T_x, T_y, T_z - translational components for the transform between the world and camera coordinate frames.

The components of the R and T matrices are called the external parameters and describe the "rigid body" affine transformation to take the real world 3-D coordinate system to the 3-D camera coordinate system. Here R is a rotation and T is a translation.

$$\vec{x} = \vec{R}\vec{X_w} + \vec{T} \tag{1.1}$$

Two further 2-D coordinate systems are defined, namely; the undistorted image coordinates (X_u, Y_u) and the the distorted image coordinates (X_d, Y_d). The two 2-D coordinate systems are related through the following equations. The first describes the perspective projection with pinhole geometry:

$$X_u = f\frac{x}{z} \tag{1.2}$$

$$Y_u = f\frac{y}{z} \tag{1.3}$$

The second set of equations relates lens distortion:

$$X_d + D_x = X_d \tag{1.4}$$

$$Y_d + D_y = Y_d \tag{1.5}$$

where (X_d, Y_d) is the true distorted image coordinate on the image plane. The coefficients D_x, D_y are:

$$D_x = X_d(\kappa_1 r^2 + \kappa_2 r^4 + \ldots) \tag{1.6}$$

$$D_y = Y_d(\kappa_1 r^2 + \kappa_2 r^4 + \ldots) \tag{1.7}$$

where $r = \sqrt{X_d^2 + Y_d^2}$. There is a further transformation from real image coordinates (X_d, Y_d) to computer image coordinates (X_f, Y_f). The following transformation is used:

$$X_f = s_x d_x'^{-1} X_d + C_x \tag{1.8}$$

$$Y_f = s_x d_y^{-1} Y_d + C_y \tag{1.9}$$

where (X_f, Y_f) is the row and column numbers of the image pixel on the camera sensor. C_x,and C_y are the row and column numbers of the centre of the sensor. The parameters d_x and d_y are the centre to centre distances in the horizontal and vertical between sensor pixel elements, that is the pixel pitch;

$$d_x'^{-1} = d_x \frac{N_{cx}}{N_{fx}} \tag{1.10}$$

N_{cx} is the number of sensor elements in the X direction. N_{fx} is the number of sensor pixel elements in the X direction as sampled by the computer, that is usable elements.

Using the Tsai algorithm there are two forms of calibration are possible, namely, coplanar - where the calibration points lie in a single plane in 3D, and non-coplanar - the calibration points occupy a 3D volume. Each case is extracted from the equations, making simplifying assumptions to reduce the number of unknowns and to get a set of overdetermined linear equations. The "solution" to this set of equations is a found as the minimizing value parameter set. The type of calibration implemented in this report is coplanar calibration. This is one of the areas to be discussed later in this report, see section 5.2, that could be improved. Another paper that discusses the above method is Ref [13], which was found to be useful.

If accurate camera calibration methods are used, then then problem of recovering depth information from stereo image pairs is simplified, see Ref [14]. In this experimental setup the authors have 2 autofocus CCD cameras, capable of optical and digital zoom. The zoom can be controlled separately for each camera. Calibration is performed on one camera at a time, as expected. However, as noted by the authors, obviously, both cameras need to be calibrated. Intrinsic parameters of the camera define the properties of the camera itself. The extrinsic parameters define the position and orientation of the camera within an arbitrarily defined 3-d space. For this experimental setup, the relative position of the two cameras is essential, as the authors argue that stereo vision becomes more complicated without it. The authors go on to say that the relative positions of the cameras is important for image rectification. The authors use the following definition: "Rectification is a transformation of the coordinate systems of the two cameras such that the image planes of the cameras are made coplanar and the scanlines in each." We will use a different definition later in this report. However image registration is defined and implemented, it must allow the unique identification of the same feature or feature set in two or more images.

Obviously getting good reference data is extremely important. In Ref [14] and Ref [20] it is stated that the accuracy of the grid is vital. From Ref [14]; "Small errors in the measurement of the pattern may lead to serious errors in calibration". From Ref [20], "The calibration pattern must be manufactured with a very high precision, since inaccuracy in the 3-D coordinates of the calibration points leads to inaccuracy in the parameter estimation".

There are many methods of camera calibration and another method examined for suitability was Ref [11]. This paper provides a discussion of the method using the correspondence of 8 points in 2 images of the same scene using calibrated cameras to reconstruct the scene. If 8 point matches are known, then the solution of a set of linear equations is involved. With more than 8 points, a least squares minimization problem must be solved. The author states that the prevailing wisdom about this method it that it is "excessively sensitive to noise". The argument in the paper is that if the data is preprocessed the algorithm can be used to give results not very different from other iterative methods.

Ref [23] presents an original approach to the problem of camera calibration using a calibration pattern. It consists of directly searching for the camera parameters that best project three-dimensional points of a calibration pattern onto intensity edges in an image of this pattern, without explicitly extracting the edges.

Ref [40] proposes a new technique to calibrate a camera. The technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. Either the camera or the planar pattern can be freely moved. The motion need not be known. Radial lens distortion is modeled. The proposed procedure consists of a closed-form solution, followed by a nonlinear refinement based on the maximum likelihood criterion. There is an informative discussion and analysis of the experimental errors involved.

There was a small problem getting the data required from the manufacturer for the particular camera model, Canon IXUS. This was not due to the data being unavailable, but due to the fact that the technical support people were not familiar with the level of detail required. The data required was the pixel pitch on the camera sensor. This is quoted in data sheets for high specification D-SLR camera models, but is not quoted regularly for compact digital cameras. See Ref [3] and other links on the site for useful information. The camera manual quotes the sensor size as being 1/1.8in. This figure is part of an arbitrary convention used to describe camera sensors, and further information is provided in the reference;

> "Sensors are often referred to with a "type" designation using imperial fractions such as 1/1.8" or 2/3" which are larger than the actual sensor diameters. The type designation harks back to a set of standard sizes given to TV camera tubes in the 50's. These sizes were typically 1/2", 2/3" etc. The size designation does not define the diagonal of the sensor area but rather the outer diameter of the long glass envelope of the tube. Engineers soon discovered that for various reasons the usable area of this imaging plane was approximately two thirds of the designated size. This designation has clearly stuck (although it should have been thrown out long ago). There appears to be no specific mathematical relationship between the diameter of the imaging circle and the sensor size, although it is always roughly two thirds."

Given the sensor in the camera used was of type "1/1.8 in", this means that the size is 7.20mm × 5.35 mm (width × height). Using this information and the number of pixels in the horizontal and vertical direction (2304×3072), the pixel pitch can be worked out for the x and y directions. For future reference, this data is hard-coded into the application. It can be externalized, but for the moment in order to effect a change in the data, a rebuild of the software is required.

Other papers were reviewed for background and further information, some of which are; Ref [24], Ref [41], Ref [28].

1.4.2 Feature recognition

The previous section examines some of the available literature describing camera calibration. All of the papers rely a process of feature extraction. There is a tacit assumption on the part of the authors describing projective geometry, that it is possible to easily, quickly and accurately identify features in a single image and their corresponding points in other images.

Some time was spent researching the following areas:

1. Edge detection
2. Line detection
3. Corner detection
4. Circle detection

5. Corner detection

Edge detection is a subject dealt with in such standard texts as Ref [10] or Ref [26]. Methods implemented for testing purposes included; Sobel edge detection Ref [10] and Laplacian of Gaussian Ref [33]. The only line detection method implemented was the Hough Transform Ref [30], Ref [31]. This can also be used to detect circles and ellipses Ref [2]. It can also be used in a more general form to detect arbitrary shapes Ref [22]. The area of corner detection is described in the following; Harris Corner Detector Ref [7] Ref [6] Ref [19], Haralick Corner DetectorRef [8], Moravec Corner Detector Ref [26], and the Beaudet Corner Detector Ref [25]. Each of the foregoing corner detectors was implemented in code. It is possible using these types of algorithms dealing with the image on a "per pixel" basis, to put them together to create more sophisticated feature recognition algorithms. This activity is explored in the following sections of this report.

Chapter 2

Project definition

2.1 Introduction

The purpose of this section is to give an overview of the how the experimental data was gathered. As the data was gathered by another team, the description will be brief, describing only the main qualities of the data collection process. It should also be noted that the data was gathered with the approval of the Faculty of Health Sciences, Internal Research Review Board, University of Ulster, which deals with research undertaken on human subjects. It was also the case that the data supplied to Brandt Technologies Ltd., was altered in such a way that the identity of the subjects was unknown.

2.2 Experiment apparatus

The experimental setup consists of the following apparatus:

1. Anatomical Rig
2. Camera (Canon Digital IXUS 750)
3. Human subject

2.3 Experiment method

The experimental team, which included an anatomist, marked particular bony landmarks on the human subjects, and placed the body markers over the area, such that the landmark lay in the centre of the body marker. Body markers were placed on the front (obverse) of the candidate and the back (reverse).

Perhaps say something like the anatomist placed body markers on anatomical locations on the torso. For the purpose of this report 6 were place on the front of the subject. The body markers are green circular "decals" or decorative stickers. For further information, please contact Justin Magee (Art and Design Research Institute) from the School of Creative Arts, University of Ulster Magee Campus, Derry.

The rig was set up in front of a blank white background. The lighting in the case of most images was fluorescent lights. The camera was setup on a tripod, such that the rig was between the camera and the blank background. The protocol describes how the subject is asked to stand, and is designed so that the posture taken up by the subject is repeatable at some later time.

The rig has a vertical axis of rotation, and can be moved with the subject resident therein. The rig can be temporarily fixed at certain angles. Photographs of the subject were taken at several stopping points, giving a set of images in which different aspects of the subject are captured.

The output of this experiment was a set of images of each subject taken with the subject holding the same relative position, while the rig is turned on its vertical axis around in a full circle.

2.4 Figures

In this section we examine figures showing the experimental setup. Figures 2.1, 2.2 and show the subject in the anatomical rig with the obverse visible. The body markers are visible on the shoulders, neck, bottom of the sternum (xiphoid) and the pelvis. The subject was standing on a black and white, square chequerboard pattern, of grid pitch 50mm. There are twelve rows of squares and eight columns. At the extremities of the chequerboard there are coloured dots; going anti-clockwise coloured; red, green, yellow and blue. Note that there are more chequerboard rows in front of the subject than are behind the subject. The purpose of the dots is to provide an orientation of the chequerboard when the image is processed. Each dot is of a different colour so that the symmetry of the board is broken. It will be seen later in the report how the dots are used as a boundary on the reference data. The chequerboard, handrails and supporting structure are placed on a turntable and the turntable pivots on a stationary base.

There is a perspex plate at the rear of the subject. This provides structural support to the rig, and is transparent so that the reverse are visible. There is another chequerboard at the top of the rig. Again this chequerboard has a grid pitch of 50mm. It has coloured dots at its corners for the same reasons as the lower chequerboard. The upper chequerboard is not completely visible in photographs of the reverse of the subject, and in particular the coloured dots are obscured by the rig itself. It is for that reason that reference data was not collected from this part of the image. As will be mentioned in section 5.2, this is an area for future research.

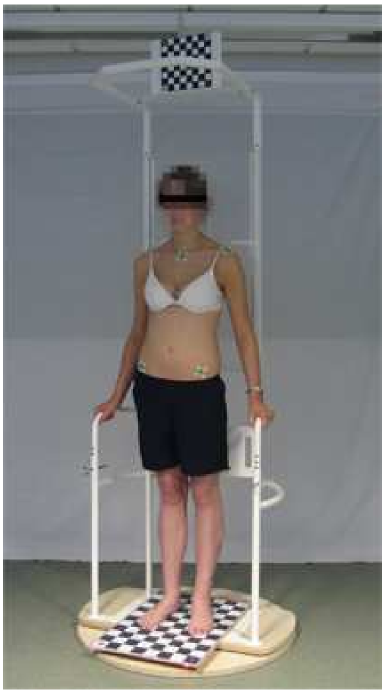

Figure 2.1: F01_02 Subject Obverse $\theta < \frac{\pi}{2}$

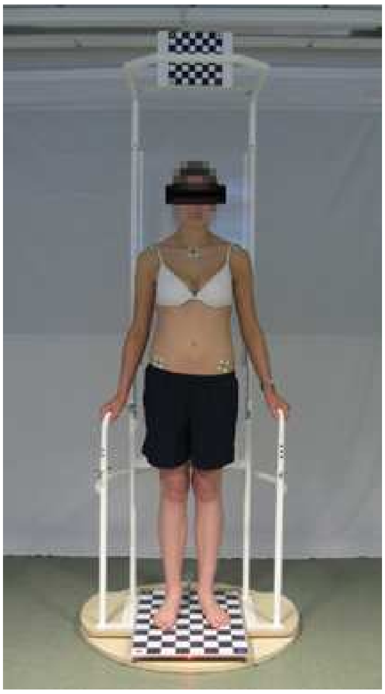

Figure 2.2: F01_01 Subject Obverse $\theta = 0$.

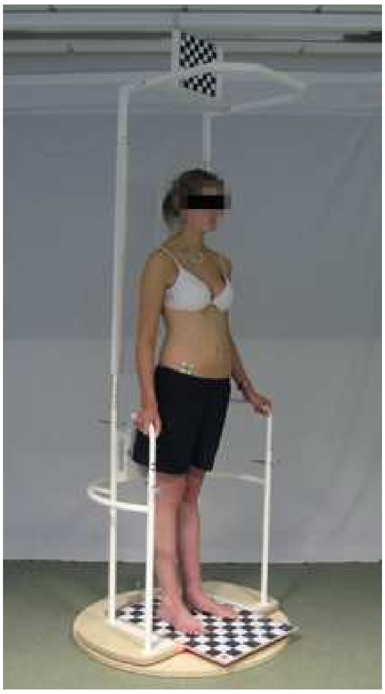

Figure 2.3: F01_15 Subject Obverse $\theta > \frac{3\pi}{2}$.

Chapter 3

Research

3.1 Introduction

In this section of the document we describe the assumptions and algorithms used to identify and locate body markers and reference data. We will describe how data are filtered to remove spurious values, leaving only data which conform to the model and will hopefully provide sensible results. We will also describe the models used to cluster data points together into a representative point in the case where there are multiple candidates for a particular feature, and the models used to cluster representative points into a (semantically) meaningful object. As with any process dealing with images, there is a hierarchy of process that takes a set of pixel data, and processes it into a, usually, smaller set of points, and further takes those points to create a, usually, smaller set of objects.

3.2 Background

3.2.1 Set theory

In this section we provide some definitions used in the construction of the binary morphological operators, see for example Ref [10]. Let Z denote the set of integer numbers: $\{0, 1, 2, 3, \ldots\}$ and let Z^2 be the set of all ordered pairs of elements (z_i, z_j), where $z_i \in Z$. The extension to Z^3 is the set of all ordered triples (z_i, z_j, z_k), where $z_i \in Z$.

Let A be a set in Z^2 and let a point in A be $a = (a_1, a_2)$, then $a \in A$.

Set complement: Is the set of all elements not contained in A and is denoted by A^c.

$$A^c = \{w \mid w \notin A\} \tag{3.1}$$

Set difference: Those elements in set A that are not in set B:

$$A \backslash B = \{w \mid w \in A, w \notin B\} \tag{3.2}$$

Set reflection: Where the set members are reflected in an axis.

$$\hat{B} = \{w \mid w = -b, \text{ for } b \in B\} \tag{3.3}$$

Set translation: Where a set is moved (translated) in the space in a given direction for a given distance. The direction and distance is specified by a point $z = (z_1, z_2)$, such that:

$$(A)_z = \{c \mid c = a + z, \text{ for } a \in A\} \tag{3.4}$$

3.2.2 Binary morphological operators

In this section we provide the definitions of the binary morphological operators, see Ref [10]. Let A and B be sets in Z^2, we define the dilation operator as follows:

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \tag{3.5}$$

An alternative description based on the definition above is given by:

$$A \oplus B = \{z \mid [(\hat{B})_z \cup A] \subseteq A\} \tag{3.6}$$

Dilation is equivalent to the Minkowski addition of two sets:

$$A \oplus B = \bigcup_{b \in B} (A)_b = \bigcup_{b \in B} (A + b) \tag{3.7}$$

Morphological dilation expands or dilates an image, meaning that it; shrinks holes enclosed by a single region and makes the gaps between different regions smaller, sometimes also denoted by $\delta_B(A)$.

We define the erosion operator as follows:

$$A \ominus B = \{z \mid (\hat{B})_z \subseteq A\} \tag{3.8}$$

An alternative description based on the definition above is given by:

$$A \ominus B = \{z \mid z + b \in A, \text{ for every } b \in B\} \tag{3.9}$$

Erosion is equivalent to:

$$A \ominus B = \bigcap_{b \in B} (A)_{-b} = \bigcap_{b \in B} (A - b) \tag{3.10}$$

Morphological erosion shrinks or erodes an image, that is, it expands the holes enclosed by a single region and makes the gaps between different regions larger, sometimes also denoted by $\varepsilon_B(A)$.

3.2.3 Preferential colour filter

Let $u : \Omega \in \mathbf{R}^2 \rightarrow \mathbf{R}^m$ be an image:

$$u : \Omega \in \mathbf{R}^2 \rightarrow \mathbf{R}^m : u = u(x, y) \tag{3.11}$$

An image can be treated as greyscale, where $m = 1$ or can be generalised a colour image (r, g, b), where $m = 3$ in the following way:

$$u : \Omega \in \mathbf{R}^2 \rightarrow \mathbf{R}^3 : u = u(x, y) \tag{3.12}$$

Define the following functions: Colour Space C:

$$C = \{(r, g, b) \mid 0 \leq r \leq 255, 0 \leq g \leq 255, 0 \leq b \leq 255\} \tag{3.13}$$

Define a colour vector, c:

$$c \in C : c = u(x, y) = (r, g, b) \tag{3.14}$$

Note that the colour c has a spatial dependence, which is implicit in this notation. For example, we could also write: $c = c(x, y)$. We may also write the colour components of c as c_i, where $i \in Z$ and $0 \leq i \leq 3$, labeling the components r, g, b. With this notation the colour preference filter is defined as follows. The image $I_{\text{red}}(x, y)$ resulting from the red colour preference filter is the set of points,

$$I_{\text{red}}(u) : \Omega \in \mathbf{R}^3 \rightarrow \mathbf{R}^3 : I_{\text{red}}(u) = \begin{cases} & r_{min} \leq \quad r \quad \leq r_{max} \\ c(x,y) & (r-g) \geq r_{\text{redLessGreen}} \\ & (r-b) \geq r_{\text{redLessBlue}} \\ \vec{0} & \end{cases} \tag{3.15}$$

where the symbol $\vec{0}$ means the RGB vector where all components are zero. Other colour preference filters are similarly defined. The green colour preference filter is $I_{\text{green}}(x, y)$:

$$I_{\text{green}}(u) : \Omega \in \mathbf{R}^3 \rightarrow \mathbf{R}^3 : I_{\text{green}}(u) = \begin{cases} & g_{min} \leq \quad g \quad \leq g_{max} \\ c(x,y) & (g-r) \geq g_{\text{greenLessRed}} \\ & (g-b) \geq g_{\text{greenLessBlue}} \\ \vec{0} & \end{cases} \tag{3.16}$$

The blue colour preference filter is $I_{\text{blue}}(x, y)$:

$$I_{\text{blue}}(u) : \Omega \in \mathbf{R}^3 \rightarrow \mathbf{R}^3 : I_{\text{blue}}(u) = \begin{cases} & b_{min} \leq \quad b \quad \leq b_{max} \\ c(x,y) & (b-r) \geq b_{\text{blueLessRed}} \\ & (b-g) \geq b_{\text{blueLessGreen}} \\ \vec{0} & \end{cases} \tag{3.17}$$

The yellow colour preference filter has to be defined differently from the previous functions, as in RGB space, yellow is not a primary colour. Yellow is made up of a combination of red and green. The yellow preferential filter is defined as follows, $I_{\text{yellow}}(x, y)$:

$$I_{\text{yellow}}(u) : \Omega \in \mathbf{R}^3 \rightarrow \mathbf{R}^3 : I_{\text{yellow}}(u) = \left\{ \begin{array}{lrcl} & y_{min} \leq & y & \leq y_{max} \\ c(x,y) & y_{min} \leq & y & \leq y_{max} \\ & (\frac{r+g}{2} - b) & \geq & b_{\text{yellowLessBlue}} \\ \vec{0} & & & \end{array} \right. \tag{3.18}$$

The filters act to suppress the background of the colours being filtered, and only pass those colours sufficiently different from their background. The sufficiency conditions are supplied by the maximum and minimum parameters in the functions. These are found by experiment and are adjustable in the software by the user.

3.3 Image segmentation of body markers

In this section we will discuss the image processing techniques used to identify the body markers. In order to locate candidates for body markers we have made the following assumptions for their characteristics:

1. Body markers are intended to be seen on any image.

2. Body markers are not obstructed by other features in an image.

3. Body markers are not occluded by the subject under test.

4. Body markers have sufficient contrast between themselves and the background feature on which they are placed.

5. Body markers will not be placed on a subject of the same or close colour to themselves.

6. Body markers do not overlap each other.

7. In RGB space the G vector has the longest component on each body marker.

8. The circular regions, G, of the body markers are connected, in a topological sense. This is the justification for dilating the image, allowing the connection of nearby clusters of pixels.

9. Body markers have a roughly square aspect ratio, in most orientations.

In fact all of the green regions of the body markers are not connected, but in practice it was not possible to use the cruciform nature of the body marker green areas. This was due to the fact that not all of the regions of most body markers were visible in all images. It is also the case that for some body markers in some images not all of the circular region was visible.

The process used to segment the body markers in any given image was the following:

1. Colour slice - application of a nonlinear filter.
2. Grayscale the image - transform based on the grayscale hue.
3. Threshold - with variable threshold.
4. Dilation - morphological nonlinear filter with variable size square structure element.
5. Label connected regions.
6. Calculate area of each connected region.
7. Calculate the centroid of each connected region.
8. Find a convex hull of 4 sides for each connected region. (Draw a box bounding the connected the region.)
9. Calculate the aspect ratio of the bounding box.
10. Filter this data based on user specified parameters of area and aspect ratio.
11. Write all of this data to a file.

The array of data containing; Area, Centroid, Bounding box and aspect ratio, defines a list of body marker candidate regions. Once we have an array containing all of the data describing each detected feature, it is possible to filter out those candidates that could not be body markers. We will use the assumptions above to derive a filter for the data. Body markers visible in an image have the following characteristics:

1. Body marker's area is less than some fixed value, upper bound; A^u .
2. Body marker's area is greater than some fixed value, lower bound; A_l.

The filter for the feature data will look something like this:

1. $Area_l \geq$ Body Marker - Area $\geq Area^u$
2. $Aspect_l \geq$ Body Marker - Aspect ratio $\geq Aspect^u$.

We will establish the fixed constants of area upper and lower bound through experiment. However, this may not be sufficiently accurate to stop spurious data being part of the data set. We assume that each subject has exactly the same arrangement of visible body markers in a given image. The images for each subject will be indexed by the "Rig Angle" θ, with θ increasing in an anticlockwise direction. For example in the first image, see figure 2.2, that is where the rig is at zero degrees $\theta = 0°$, each subject has the following visible body markers:

1. Shoulder left (SL)
2. Shoulder right (SR)
3. Neck (N)
4. Sternum (S)
5. Pelvis left (PL)
6. Pelvis right (PR)

where each marker such as N is a point in the two dimensional space $\Omega \in \mathbf{R}^2$. In the next section we will describe how they are configured.

3.3.1 Image segmentation - Object model assumptions

For the purposes of this paragraph, we are considering only one body marker model, that is the obverse model present in those images for which the rig is at zero degrees $\theta = 0°$. The object we are going to describe in detail is the arrangement of body markers. The relationship of each body marker to its neighbour defines the object. Relationships between points are defined in terms of a point being either to the left or right, or above or below, each other, or at least above a certain distance away . This last criterion is to exclude points that are too close together. The smallest distance at which points are considered separate is called the noise threshold. The main characteristic of the object models described in this report and implemented in the code is that, apart from the noise threshold, there is not other length scale used. This allows a general model to be described that will fit all "reasonably" shaped human candidates. We start to define the obverse body model using the following assumptions :

1. The left SL and right SR shoulders and sternum S markers lie on a circle centred on the neck, N.
2. The left (PL) and right (PR) pelvis markers lie on a circle centred on the sternum, (S).

The assumptions are based on the symmetry of human anatomy. The first assumption describes the upper body circle. The set of relationships in this statement describe the relative positions of the neck (N), left and right shoulders SL and SR respectively. Consider a circle centred on the neck, (N), with radius $r = d(N, S)$. This circle will also go through the sternum S and the left and right shoulders, SL and SR respectively.

$$d(N, SL) = d(N, SR) \tag{3.19}$$
$$d(N, SL) = d(N, S) \tag{3.20}$$

In specifying these relations we have used the Euclidean distance metric $d(a, b)$, of the point a to the point b. The second assumption specifies the relation of the pelvis markers to the sternum. In this case we have:

$$d(S, PL) = d(S, PR) \tag{3.21}$$

In practice there will rarely be exact equality in the above relations because of the variation of human posture, so they are replaced for the purposes of putting them into code, by the following inequalities. We are using empirically defined tolerances, noise thresholds, for the upper body circle, tol_u and the lower body circle, tol_l:

$$\begin{aligned} \mid d(N, SL) - d(N, SR) \mid &\leq tol_u \qquad (3.22) \\ \mid d(N, SL) - d(N, S) \mid &\leq tol_u \\ \mid d(S, PL) - d(S, PR) \mid &\leq tol_l \end{aligned}$$

We now need to specify spatial relations between the body markers. In addition to 3.22, the following obvious statements suffice to place the body markers in relation to each other:

1. The left shoulder is left of the neck and left of the right shoulder.
2. The right shoulder is right of the neck and right of the right shoulder.
3. The neck is above the sternum.
4. The left pelvis marker is left of the sternum.

5. The right pelvis marker is right of the sternum.

It is not wise to say that the shoulder markers are above the neck as this could easily be violated. The assumptions are made to cover the simplest geometry of human anatomy. As above, we now describe the assumptions made above in quantitative terms.

$$
\begin{align}
N.x &> SL.x \tag{3.23}\\
N.x &< SR.x \tag{3.24}\\
N.y &> S.y \tag{3.25}\\
N.y &> S.y \tag{3.26}\\
S.x &> PL.x \tag{3.27}\\
S.y &> PL.y \tag{3.28}\\
S.x &< PR.x \tag{3.29}\\
S.y &> PL.y \tag{3.30}\\
& \tag{3.31}
\end{align}
$$

Where we have used the notation that $.x$ and $.y$ means the x and y coordinates of whatever precedes them. See Figure 3.1

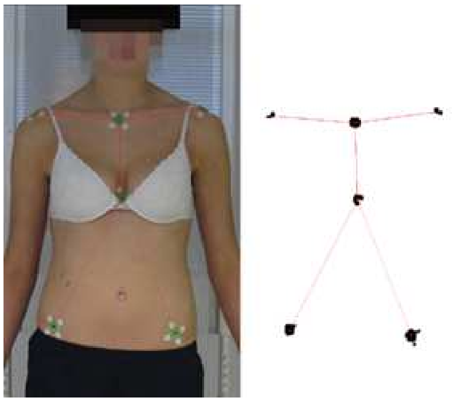

Figure 3.1: Obverse body model F01_01 Subject Obverse $\theta = 0$.

The reverse body model is defined similarly. There are ten body markers on the reverse of the candidate. They are labeled for our purposes as follows:

1. Neck 1 ($N1$)

2. Neck 2 ($N2$)

3. Neck 3 ($N3$)

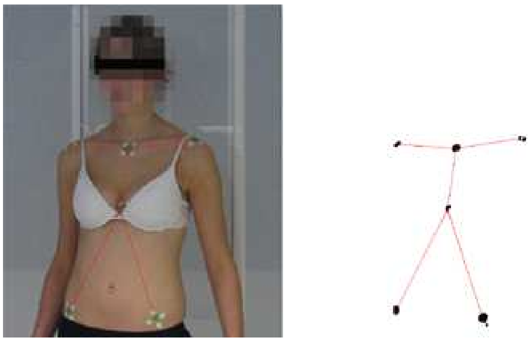

Figure 3.2: Obverse body model F01_01 Subject Obverse $\theta < \frac{\pi}{2}$.

4. Shoulder Left (SL)
5. Shoulder Right (SL)
6. Mid back (MB)
7. Lower back 1 ($LB1$)
8. Lower back 2 ($LB2$)
9. Pelvis Left (PL)
10. Pelvis Right (PR)

The assumptions making the model are as follows:

1. The neck markers are equidistant in the vertical; $\mid (N1.y - N2.y) - (N2.y - N3.y) \mid \leq noiseThreshold$
2. $N1.y > N2.y$
3. $N2.y > N3.y$
4. By symmetry $\mid d(N3, SL) - d(N3, SR) \mid \leq noiseThreshold$
5. By symmetry $\mid d(MB, SL) - d(MB, SR) \mid \leq noiseThreshold$
6. $SL.x < SR.x$
7. $SL.x < MB.x$ and $MB.x < SR.x$
8. By symmetry $\mid d(LB1, PL) - d(LB1, PR) \mid \leq noiseThreshold$
9. $LB1.y > LB2.y$

10. $LB2.y > PL.y$

11. $LB2.y > PR.y$

12. By symmetry $\mid d(LB2, PL) - d(LB2, PR) \mid \leq noiseThreshold$

13. $PL.x < LB2.x$ and $LB2.x < PR.x$

It is the case that in images for $\theta < \pi$ and $\theta > \pi$ that not all of the body markers are visible. For these cases there are two further models based on the above, each with nine body markers. In each reverse model the left shoulder SL and right shoulder SR is removed. The reverse images show the three types of object model required to describe the body markers when one is occluded. It would be possible to describe situations where more than one is occluded, but there is a minimum number of body markers required to describe the model. We have done no work on finding the minimum number. The minimum number of body markers must exclude the possibility of noise aliasing.

What we are trying to define is a pattern of points for which we can search. The idea is similar to pose estimation. In our case we only know qualitative features of the structure being posed. It is hoped that because the characteristics of groups of features will be robust to noise, and can be used to identify the same structure in subsequent images, where the angle of the rig is different from zero.

3.4 Image segmentation to identify calibration data

In this section we will discuss the assumptions and model describing the calibration data. The chequerboard at the subjects' feet contains the calibration data will use. A detailed description of the chequerboard was given in a previous section. The process used to identify the reference data is as follows:

1. Find the chequerboard coloured dots.

2. Coloured dots are used to define a quadrilateral.

3. This quadrilateral is projected using perspective projection Ref [9], Ref [39] onto a reference grid square.

4. Apply the Harris corner detector to the image.

5. Filter the corners.

6. Apply the Jordan Curve Theorem Ref [15] Ref [32] Ref [4] to identify those points inside the quadrilateral defined by the coloured dots.

7. The perspective projection defined by taking the coloured dots quadrilateral to the reference square is applied to the filtered corner data so that they are mapped to within the reference square. The coloured dots maintain the orientation of the grid, so that this process is exactly the same when applied to the grid viewed from the reverse.

8. It is assumed that the projected data will be close to a grid with horizontal and vertical lines. The perspective of the corner data in the image has been removed by this transformation.

9. Histograms of the projected data are taken in the horizontal and vertical directions to find the statistical width and height of the grid from the projected data.

10. From this the grid pitch is found, that is the horizontal and vertical distance between rows and columns is found. This and the width and height define the grid.

11. Filter the projected corner data again based on distance from the grid.

12. The projected data closest to the grid are assigned world coordinates based on the grid position.

13. The corner data image coordinates, projected data coordinates and world coordinates are saved to a file.

There is an important point related to accuracy here. If the coloured dots do not correspond closely to a regular square, reference data will be ignored. An important point to note is that the positions of the reference data are not affected by the found positions of the coloured dots, but if the grid is not close to the vertical and horizontal, then reference data will be filtered out. The projection of corner data onto the a reference square is used as the basis for registering two or more images. This will be discussed later in the report.

Figure 3.3 shows the chequerboard before any processing takes place.

Figure 3.4 shows the chequerboard obverse model found and highlighted in the image.

Figure 3.5 shows the chequerboard image with the Harris corner detection data back projected onto the image. It can be seen that each corner candidate

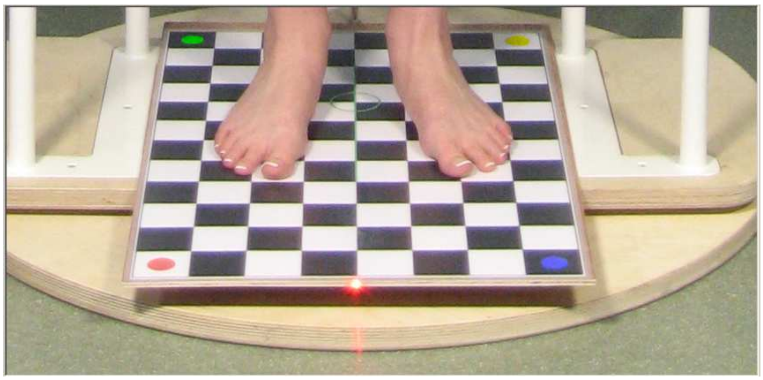

Figure 3.3: Chequerboard before any processing.

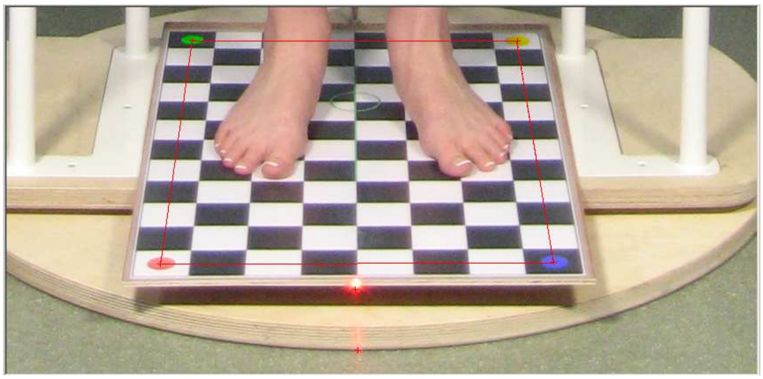

Figure 3.4: Chequerboard obverse model $\theta = 0$.

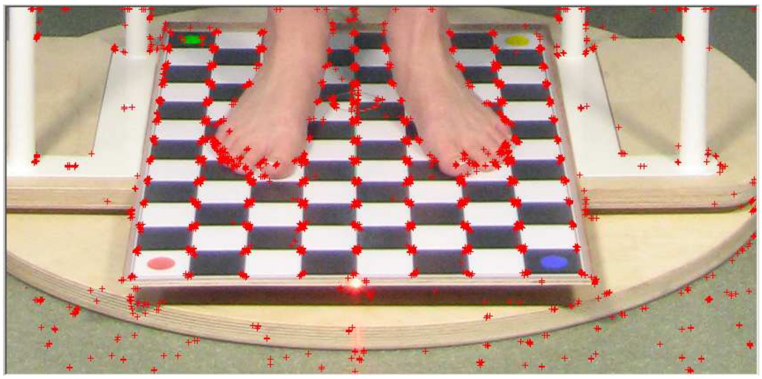

Figure 3.5: Unfiltered Harris corner detection back projected onto the chequerboard image.

has many points. It can be seen that there are spurious corners found, and many valid corners that do not correspond to reference data.

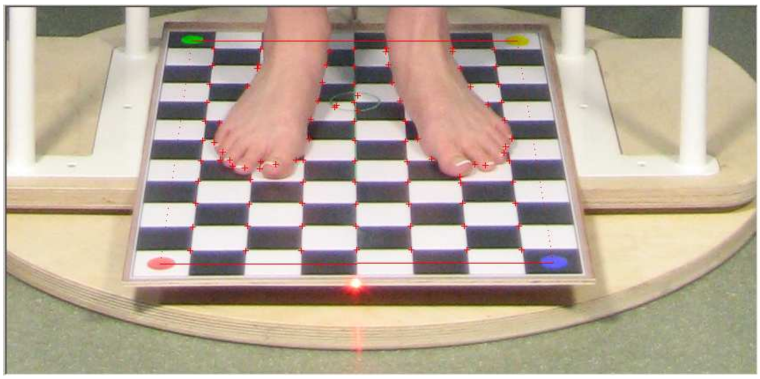

Figure 3.6: Filtered Harris corner detection back projected onto the chequerboard image inside the object model.

Figure 3.6 shows the chequerboard image with the filtered Harris corner detection data and obverse chequerboard model back projected onto the image. The corner data has been filtered to give a better representation of corners. There is still some data that represents valid corners, but does not represent valid reference data.

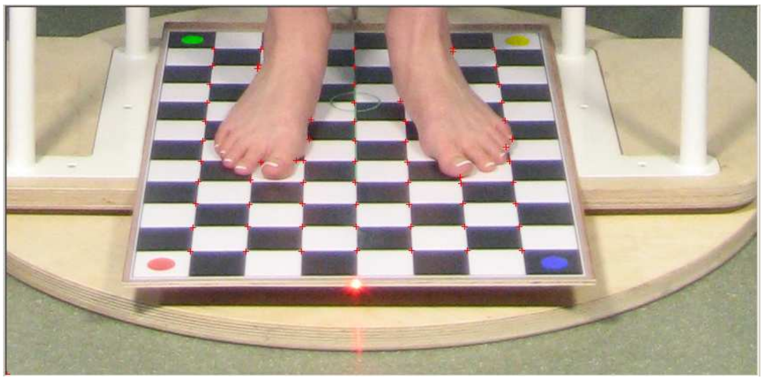

Figure 3.7: Filtered and rectified Harris corner detection back projected onto the chequerboard image inside the object model.

Figure 3.7 shows the chequerboard image with the filtered and rectified Harris corner detection data and back projected onto the image.

Figure 3.8 shows two images. The image on the left hand side shows all of the corner data projected onto the reference grid square. As described above, histograms taken in the horizontal and vertical directions are used to

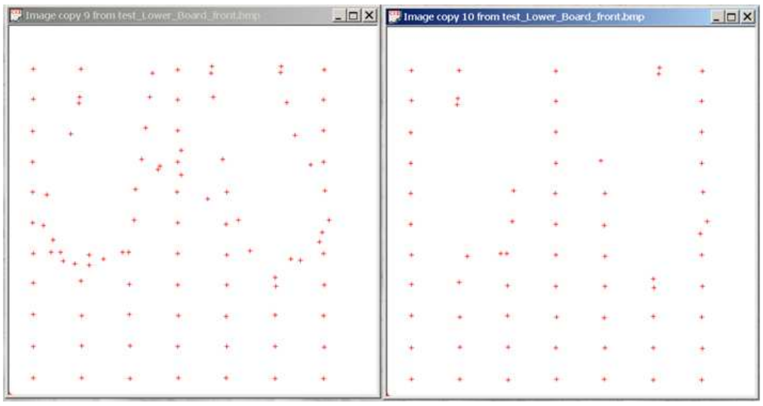
Image copy 9 from test_Lower_Board_front.bmp
Image copy 10 from test_Lower_Board_front.bmp

Figure 3.8: Left hand shows unfiltered data. Right hand side shows data filtered on distance from the grid.

determine the height and width of the grid and grid pitch. This data is then used to define the grid. The left hand side image shows the projected corner data filtered on its distance from the grid. This distance can be set by the user of the software.

3.4.1 Corner Detection

Corner detection is a very important component of this project. The actual implementation of the corner detection algorithm loosely follows the outline given in Ref [5]:

1. Open an image.
2. Create a copy of the original image and blur it with a Gaussian filter.
3. Apply the Sobel horizontal edge detector.
4. Create a copy of the original image and blur it with a Gaussian filter.
5. Apply the Sobel vertical edge detector.
6. Calculate the autocorrelation matrix, M, for each point.
7. Calculate det$M$. This is the product of the eigenvalues of the matrix.
8. Calculate Trace$M$. This is the sum of the eigenvalues of the matrix.
9. Define a map $= det(M) - k(trace(M))^2$, where k is the Harris constant.
10. Threshold the map.

11. Filter the points on the map.
12. Those non-zero points left in the map are corners.

The Harris constant and threshold can be set in the software.

3.5 Camera calibration

The area of camera calibration was discussed in detail in a previous section. In this section we will discuss camera calibration as it is relevant to the software. Camera calibration is the process of finding the parameters of transformations that take real world coordinates and map them to image coordinates. The problem statement taken from Ref [29] is; "to compute the camera extrinsic and intrinsic parameters based on a number of points whose object coordinates (world coordinates) in the (x_w, yw, zw) coordinate system are known and whose image coordinates (X, Y) are measured". The reference data are used in the code from Ref [38] to produce a camera model. Two images are used to create camera models. There is a function which calculates the image coordinates based on world coordinates as input. Using the body marker data from two images as image data, it is possible to define a function whose minimum is one set of world coordinates that should give the position of the body markers in 3-D coordinates. The code that implements this is loosely based on Ref [18].

3.6 Image registration

In this section we discuss image registration. See Ref [42] for the following definition of image registration: "Image registration - the process of overlaying two or more images of the same scene acquired from different viewpoints, by different sensors and/or at different times so that the pixels of the same coordinates in the images correspond to the same part of the scene." There are at least two approaches to registration; intensity driven approaches and landmark based approaches Ref [17]. Given how the experiment was set up, we felt that the best approach to registering two or more images was landmark based registration. In the case of the experimental setup we have;

1. the camera location is fixed.
2. the human subject is fixed in the rig.
3. the rig turns about a vertical axis.

4. the remainder of the scene is stationary.

For intensity based approaches, the entire scene remains stationary, and the camera moves. Our problem is that significant elements within the scene have changed position. We have to extricate a segment of each sequence of images which can be used in an image registration algorithm. Image registration for our problem became straightforward when we used the object model. The coordinates of each point in the model directly correspond with each element, such as neck, pelvis and so on. The same is true of the chequerboard. Once the coordinates have been mapped to the standard square and provided with a grid, then the mapping between images is trivial, even if there is a large distortion.

The area of mapping a cloud of points to another, or registering one cloud of points to another is an open problem in the literature, Ref [1], Ref [16], Ref [12] . Even in our 2-D problem of mapping a set of points to a grid, minimizing the sum of the distances of all points to a grid could be an independent problem by itself. The problem statement would have to include the fact that the grid parameters are also being optimized based on the point cloud data.

Chapter 4

Software implementation

4.1 Introduction

In this sections we discuss how to use the software to produce data. We also discuss the process that should be used to identify the set of parameters best suited to the image.

4.2 Body marker image segmentation

The first thing that should be identified in the image is the body marker configuration; obverse or reverse. The obverse is visible if all of the six body markers are visible on the front of the candidate. This section describes how to find body markers only. There is no object model fitting involved.

Results showing unfiltered body marker data can be seen in figure 4.1 Information on the *RGB* value of the pixel underneath the mouse can be seen in the bottom left hand corner of the application. The preferential filter can be improved to remove noise by changing the body marker area, aspect ratio and colour parameters. It can be seen that the body marker candidates are back projected in the form of red crosses on the original image. This indicates likely positions for the body marker locations.

The menu items using the parameters from ”body marker parameters” include 3CEM | Process body markers | Obverse, 3CEM | Process body markers | Reverse and the menu items under 3CEM | World Coordinate calculation. The user should experiment with one image and vary the parameters to see the effect. If a body model is not found, then a dialog will appear to tell the user. The results for this process are show in figure 4.2

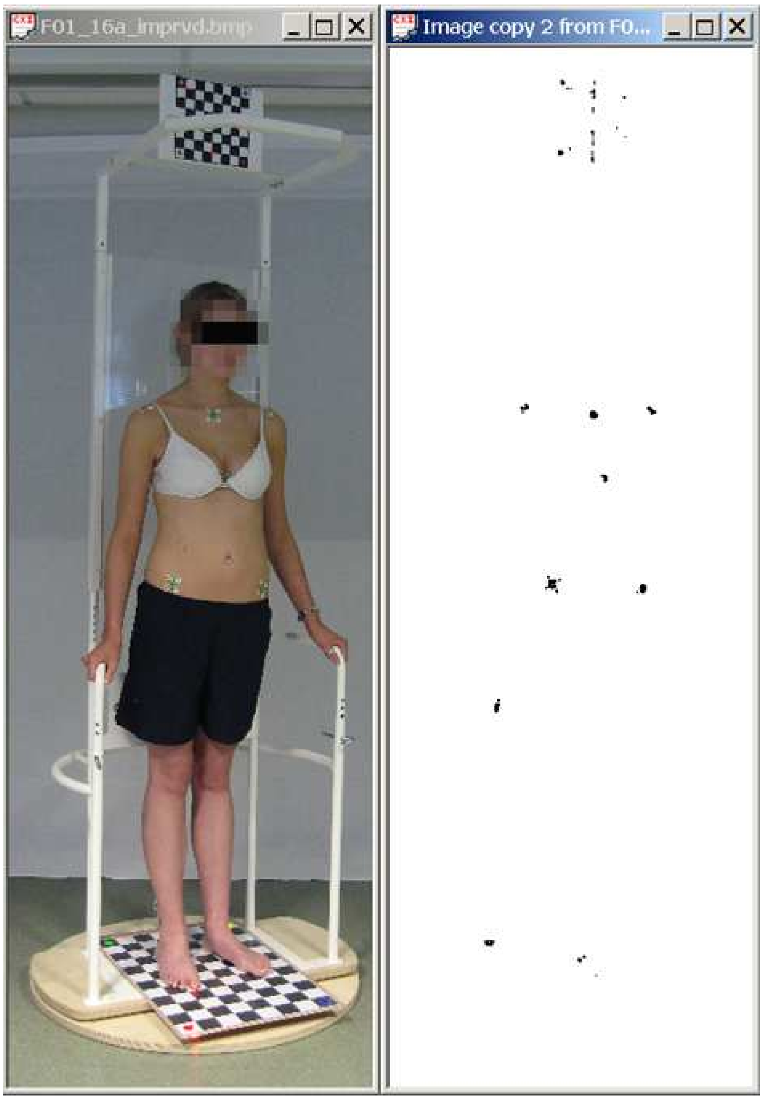

Figure 4.1: Results of body marker segmentation, showing all unfiltered data.

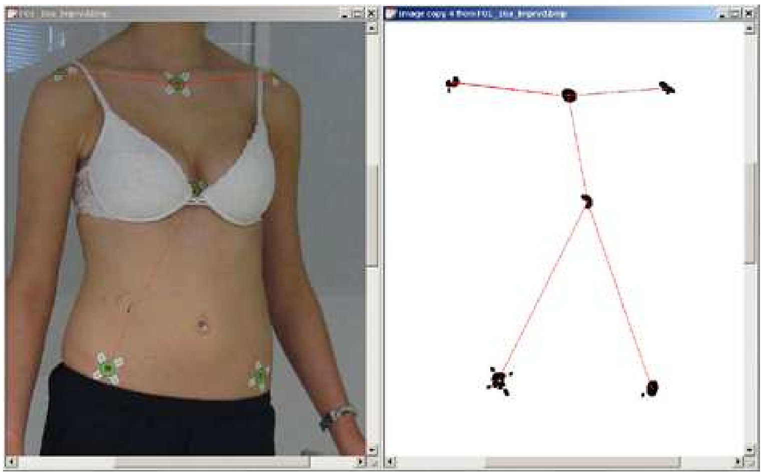

Figure 4.2: Results showing the obverse body model.

4.3 Camera calibration data

The process of finding the world coordinate data depends on having a set of configuration data that will allow the software to find the object models and the reference data in an image. Without this data set correctly the process will fail prompting the user with a dialog box stating which object model cannot be found and to start the process again with different parameters.

4.4 World Coordinate data

We have seen in the previous sections of this chapter how to set and find the parameters necessary locating the object models in images, see for example section 4.2. In this section we will explain how to use the calibration data and body marker data generated in section 4.3 to find the world coordinates of the body markers. The program requires the data from two images to calculate the world coordinates.

4.5 Results

In this section we present some data generated by the program. The data is presented for the obverse model only, for reasons of time pressure. The data was calculated from ”improved” and cropped graphics, and from a scientific point of view, the data is only provided for illustrative purposes only. It should be noted that the cropped images means that from the calculation point of view, the image is on the left hand side of the image plane. The data was cropped to improve speed of testing. Images were improved in the following sense; where a body marker was visible, but was practically gray in *RGB* value, that body marker was given a green colour. Where a body marker was not visible, as in for example F10, the data was not used. There were also improvements made to the coloured dots on the chequerboard, these changes consisted of giving the dots a stronger colour. This had no effect on the location of the reference data, and is equivalent to user intervention in the process to identify the coloured dots as landmarks.

4.6 Error analysis discussion

A full analysis of the errors in the process is a non-trivial task. For the moment we will discuss only sources of error, which include and are not limited to:

1. Body marker location error systematic error due to image processing.
2. The corner data not being accurately placed due to the potential anisotropic response of the corner detector.
3. Need to examine if there is a systematic difference between the corners as perceived by the human user and those points located by the corner detection process.
4. Algorithm errors in treating the reference data - coplanar data, non-coplanar data.
5. Accuracy of camera parameters from manufacturer.
6. Systematic error due to camera resolution.
7. Systematic error due to rig - The turntable has to remain horizontal at all times while taking an exposure - no error due to motion blur of the subject.
8. Systematic error due to subject not staying perfectly still between photographs.

The first three points in the above list are potentially the biggest contributors to the error in the current process. We feel that in order to improve the process, these sources of error need to be quantified, and assessed in relation to the data to provide a proper estimate of error. This information will provide a context to understand the results presented in the tables above, and may provide information that will help in automating the process of 3-D coordinate extraction.

Chapter 5

Conclusions and future research

5.1 Introduction

In this section we examine some of the areas that could be improved upon in the project, and discuss some areas for future research.

5.2 Particular improvements

Some of the ways in which the errors in the data points could be reduced are as follows:

1. Non coplanar camera calibration, Ref [29], could be implemented.

2. The implementation of Tsai's Ref [29], error calculation.

3. There is a pixel error associated with the output measurements made. This should be converted to a millimeter measurement.

4. The corner detector might not have an isotropic response.

As mentioned in a previous section, there are several potential sources of error in gathering the data, some of which are as follows;

1. Moving either the rig or camera tripod during the experiment, between photographs. As long as the relative positions of the camera and rig are maintained for one subject, while the data is being captured, that is sufficient.

2. The human subject changing position during the data capture either from fatigue or due to movement of the rig will introduce error.

3. The number of squares of the grid visible in the reverse view of the subject is smaller than the number visible in the obverse view. This has an impact on the quality of the data.

The software implementation could be improved in many ways to reduce error, some of which include;

1. The biggest improvement that could be made to the process would be to include the calibration data from the upper chequerboard.

2. Isotropic edge detection input to the corner detector.

3. Subpixel accuracy for corner location could be applied.

4. The image could be processed only in the area in which the chequerboard model is found improving efficiency.

5. An automated test could be used to test the calibration data being generated to see if the grid is correct.

Bibliography

[1] Alexander Patterson Ameesh Makadia and Kostas Daniilidis. Fully automatic registration of 3d point clouds. *To appear in IEEE Conference on Computer Vision and Pattern Recognition.*

[2] Pascal Ballester. Applications of the hough transform. http://cadcwww.dao.nrc.ca /ADASS/adass_proc/adass3/papers/ballesterp/ballesterp.html. Web article.

[3] Vincent Bockaert. Sensor type designation. http://www.dpreview.com /learn/?/Glossary/Camera_System/Sensor_Sizes_01.htm. Web article.

[4] Octavian Cismasu. The jordan curve theorem for polygons. http://www-cgrl.cs.mcgill.ca /godfried/teaching/cgprojects/97/Octavian/compgeom.html. Web article.

[5] Harris corner detector. Corner detectors. http://www.cim.mcgill.ca/ dparks/CornerDetector/harris.htm. Web article.

[6] Dmitrij Csetverikov. Basic algorithms for digital image analysis: A course. http://ssip2003.info.uvt.ro/lectures/chetverikov/corner_detection.pdf. Web article.

[7] Konstantinos G. Derpanis. The harris corner detector. http://www.cse.yorku.ca /kosta/CompVis_Notes/harris_detector.pdf. Web article.

[8] Paulo Dias. Haralick corner detector. http://homepages.inf.ed.ac.uk /rbf/CVonline/LOCAL_COPIES/DIAS2/. Web article.

[9] David Eberly. Perspective mapping between two convex quadrilaterals. http://www.geometrictools.com/Documentation/PerspectiveMapping.pdf. Web article.

[10] Richard E. Woods Gonzalez, Rafael C. *Digital Image processing*. Prentice Hall, 2002.

[11] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997.

[12] Frank Hoffmann, Klaus Kriegel, Sven Schnherr, and Carola Wenk. A simple and robust geometric algorithm for landmark registration in computer assisted neurosurgery.

[13] Berthold K.P. Horn. Tsai's camera calibration method revisited. http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf. Web article.

[14] Jo Abrantes Michael Tapper, Phillip McKerrow. Problems encountered in the implementation of tsai's algorithm for camera calibration. In *Proceedings 2002 Australasian Conference on Robotics and Automation*, pages 66–70. Australasian Conference on Robotics & Automation, 2002.

[15] Emil G. Milewski. *The Topology Problem Solver*. REA, New Jersey, 1998.

[16] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas. Registration of point cloud data from a geometric optimization perspective. In *Symposium on Geometry Processing*, pages 23–31, 2004.

[17] Jan Modersitzki. *Numerical Methods for Image Registration*. Oxford University Press, New York, 2002.

[18] Chris Needham. Calibration. http://www.comp.leeds.ac.uk/chrisn/Tsai/. Web article.

[19] Donovan Parks and Jean-Philippe Gravel. Corner detectors. http://www.cim.mcgill.ca/ dparks/CornerDetector/index.htm. Web article.

[20] Hcm Network Participation. Vision algorithms and optical computer architectures.

[21] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing.* Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

[22] Philipp Robbel. Pose estimation using the hough transform. http://homepages.inf.ed.ac.uk
/rbf/CVonline/LOCAL
_COPIES/AV0405/ROBBEL/index.html. Web article.

[23] Luc Robert. Camera calibration without feature extraction. *Computer Vision and Image Understanding: CVIU*, 63(2):314–325, 1996.

[24] Mubarak Shah. Fundamentals of computer vision. http://www.cs.ucf.edu/courses/cap6411/book.pdf. Web article.

[25] Eduard Sojka. Pose estimation using the hough transform. http://www.cs.vsb.cz/sojka/cordet/presentation.html. Web article.

[26] V. Hlavac Sonka M. and R. Boyle. *Image Processing, analysis, and machine vision.* PWS Publishing, Pacific Grove, CA, 2nd ed., 1999.

[27] Jonathan Thornburg. Numerical programming resources. http://www.aei.mpg.de/ jthorn/numerical.html. Web article.

[28] Miroslav Trajkovic and Mark Hedley. Rigid motion recovery from less than eight feature point matches. http://www.bmva.ac.uk/bmvc/1997/papers/083/083.html. Web article.

[29] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J. Robotics Automat.*, RA-3(4):323–344, Aug. 1987.

[30] Unknown. Hough transform. http://www.homepages.inf.ed.ac.uk
/rbf/HIPR2/hough.htm. Web article.

[31] Unknown. Hough transform. en.wikipedia.org/wiki/Hough_transform. Web article.

[32] Unknown. Jordan curve theorem and its generalizations. http://www.math.ohio-state.edu
/fiedorow/math655/Jordan.html. Web article.

[33] Unknown. Laplacian/laplacian of gaussian. http://www.cee.hw.ac.uk/hipr/html/log.html. Web article.

[34] William T. Vetterling, William H. Press, Saul A. Teukolsky, and Brian P. Flannery. *Numerical recipes: example book (C)*. Cambridge University Press, New York, NY, USA, 1988.

[35] Jiri Walder. An experimental system for reconstructing a scene from the image sequences produced by a moving camera. http://www.cg.tuwien.ac.at /studentwork/CESCG/CESCG99/JWalder/paper.pdf. Web article.

[36] J. Weijer, T. Gevers, and A. Smeulders. Robust photometric invariant features from the color tensor, 2004.

[37] James R. Williamson. What is photogrammetry? http://www.123photogrammetry.com/photogrammetry.html. Web article.

[38] Reg Wilson. Tsai camera calibration software. http://www.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html. Web article.

[39] Christopher R. Wren. Perspective transform estimation. http://alumni.media.mit.edu/ cwren/interpolator/. Web article.

[40] Zhengyou Zhang. A flexible new technique for camera calibration. http://research.microsoft.com/ zhang/Papers/TR98-71.pdf. Web article.

[41] Zhengyou Zhang, Rachid Deriche, Olivier D. Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995.

[42] Barbara Zitova. Image registration of blurred satellite images. http://staff.utia.cas.cz/zitova/registration.htm. Web article.