Insert Title Here :P

Your Mother

February 8, 2013

Chapter 1

Specification

1.1 Problem Statement

Computer Vision and object recognition are becoming more prevalent in day-to-day life, entering the home through commodity hardware such as the Microsoft Kinect. Because of this, many fields are looking to use the Kinect to automate or improve many of their processes. One such field is medicine, where medical imaging, specifically range imaging is rapidly developing in a number of different contexts, such as patient setup, and being applied in a variety of situations. Advancements in this area are leading to better patient monitoring, diagnosis and treatment of patients in various sub domains in Medicine.

The PARSE project aims to create an application toolkit to increase the reliability and accuracy of the measurements taken when monitoring weight loss and body size variations of a patient. The toolkit will provide an easy to use interface allowing for medical personnel to take measurements of body volume, limb circumferences and relative positions of ultrasound scanners on the surface of the body. These measurements can be stored persistently and be recalled later as part of a patients medical record. The remit for achieving this will be in our use of commodity hardware - specifically the Microsoft Kinect.

1.2 Motivation

The motivation for this project came from several limitations associated with cost and the accessibility of current practice for measuring weight loss, body shape and the calibration/configuration of non-invasive medical scanning equipment. Body volume is typically measured using multiple sensors [?] or expensive air displacement plethysmography equipment [?]. A need was identified for a system that could accurately measure a patient's total body volume using a single piece of commodity hardware such as the Kinect.

The traditional method of measuring circumference of body parts such as arms or torso is limited by the accuracy of the physical device used to measure it, usually a tape measure. A bigger problem, however, arises when taking multiple readings. For the measurements to be useful the exact same point must be measured each time. A need was stated for the new toolkit to accurately identify the circumference of individual body parts, either from the original body scan or a new individual scan of the limb, and to enable the registration of and consistent guidance to the point of measurement.

The depth of subcutaneous fat in a person can also be measured to determine a patient's weight loss or variation in body size, and the depth of the fat is also an indicator of other serious conditions such as insulin resistance [?] and coronary heart disease [?]. The problem here is measuring the depth of this subcutaneous fat consistently by placing the ultrasound scanner in precisely the same place, at the same angle, when taking multiple readings of the same body part over time. If this is not the case, errors can be introduced and may give a false indicator of loss or gain in subcutaneous fat. The PARSE team feel that the proposed toolkit can aid this, by recording the position of the first scan and providing direction to the user for subsequent scans. This guided positioning will be achieved using a mixture of the Kinect skeletal tracking and object recognition algorithms on the depth and image feeds from the Kinect scanner.

1.3 Requirements

1.3.1 Functional

someone please may it as sexy as the sexy table in the spec

1.3.2 Non Functional

someone please may it as sexy as the sexy table in the spec

1.4 Point Cloud Stitching

As only one scanner is to be used it is necessary to produce multiple scans so that the patient's whole body is scanned. Each of these scans must then be stitched together to form one big model which can then be manipulated as necessary.

1.4.1 Surface Normal Calculation

The majority of the algorithms pertaining to point cloud stitching require the surface normal field to be calculated. This would present a challenge if we were to estimate the normal to each point in each scan as it would simply require too many calculations to be made. As a result it was decided that some sampling method should be used. In this case a voxel grid filter was implemented to create a sparse representation of the object that had just been scanned.

Chapter 2

Research

Chapter 3

Design

This Chapter will describe the design choices made by the group.

3.1 Person Isolation

This section details the creation of a method to isolate a person from the background data. An advantage of using a Kinect over another camera is that the Kinect is able to isolate a skeleton associated with a person. This skeleton meant that the group did not have to make use of computationally heavy computer vision algorithms to isolate a person . example

Instead, the method developed used the skeleton to determine the approximate depth of the person (using the HipCenter joint depth). Any point whose depth value is outside a delta of the HipCenter's depth is discarded.

Cutting off base depth alone is not enough, as this method leaves a *ring* of equal distant points in-line with the person. To eliminate this ring, the positions of left and right most point of the person (i.e. HandLeft and HandRight) are calculated and anything outside of this range is also discarded.

Section **??** details the effectiveness of this approach.

3.2 Volume Estimation

This section will focus on the algorithm design for the Volume Estimation module.

3.2.1 An Initial Upper Bound Approximation

It was decided as a 1st approximation to calculating the *minimum bounding box* of a person, and use this box to determine volume. A minimum bounding box is defined as follows in **??**.

Definition: For a point set in N dimensions, it refers to the box with the smallest measure (area, volume, or hypervolume in higher dimensions) within which all the points lie.

Figure 3.1: Minimum Bounding Box Definition [**?**]

For the purposes of PARSE, this box is three dimensional.

The PointCloud class design described in was amended to keep track of the minimum and maximum x, y and z co-ordinates of all the points stored. From this the volume of the minimum bounding box can be calculated as in Figure **??**.

cite section

$$Volume = (xmax - xmin) * (ymax - ymin) * (zmax - zmin)$$

Figure 3.2: Calculating the Volume of the Minimum Bounding Box

This gives the volume of the minimum bounding box in point cloud Space, rather than in three dimensional reality. A multiplier must be applied to the volume calculated by Figure **??** to facilitate a transform between the two. The precise value of this multiplicative can only be determined through testing, see section .

cite

Algorithm Running Time

Any algorithm for calculating volume will dealing with large data sets where the number of points p in the point cloud is in the order of 10^6. As such, the algorithms computation complexity must be taken into account.

This algorithm runs in $O(1)$ in terms of p as *xmax* et al are stored within the point cloud object upon creation and can be retrieved in constant time.

3.2.2 A Refined Approach

The method described above in Section **??** will result in an overestimation of a person's volume. This overestimation is due to the fact people are not rectangular. The refined approach is based on the technique of *volume rendering*. Volume rendering uses multiple two dimensional slices to build up a three dimensional image. In the case of PARSE, the area of these two dimensional slices is calculated and summed to determine the volume of the three dimensional point cloud. These areas may need to by multiplied by a constant in order for an accurate volume to be returned. This multiplicative can only be determined through testing, see Section . cite

The area of the two dimensional slices is calculated using the Shoelace formula [**?**]. The Shoelace formula states that given a Polygon P made up of points $\{(x_1, y_1), (x_2, y_2)...(x_n, y_n)\}$, the area can be calculated as in Figure **??**.

$$Area = \left|\frac{(x_1y_2 - x_2y_1) + (x_2y_3 - x_3y_2) + ... + (x_ny_1 - x_1y_n)}{2}\right|$$

Figure 3.3: The Shoelace Formula [**?**]

Algorithm Running Time

The Volume Rendering runs in $O(z)$ where z is the number of points in the z axis. z is itself bounded by $O(p)$, where p is the number of points in the point cloud.

The *getAllPointsAt* and the *rotSort* method occur within the volume rendering for loop. The *getAllPointsAt* method runs in $O(f)$. check method name

The *rotSort* method is broken down into several parts. The first part is to determine the approximate centre of the point set. This calculation involves iterating through the list of points in the given plane, which is bounded by $O(p)$. check The points then need to be translated so that the center is aligned with the origin, again requiring $O(p)$ time. The sorting of the points (using quicksort) takes $O(p * log(p))$ in the average case and $O(p^2)$ in the worst case. The points are then translated back to there original position, again requiring $O(p)$ time. Once the sorting has been completed, the volume is calculated using the Shoelace formula described above, requiring $O(p)$.

Hence the contents of the volume rendering for loop is bounded by $O(p * log(p))$ in the average case. Meaning the entire algorithm requires $O(p^2 * log(p))$ in the average case and $O(p^3)$ in the worst case. or O(f) find f

These bounds are in no way a proven *tight* upperbound for the algorithm, as there have been many simplifying assumptions. Such as $O(z)$ growing as fast as $O(p)$, whereas in reality, $O(z)$ is will grow slower than $O(p)$, perhaps by an order of magnitude.

For quicksort, the worst case occurs when the algorithm consistently picks the worst pivot each iteration. The probability of this happening is $O(\frac{1}{p^2})$, as such it is unlikely to occur in general use. In the case of PARSE, p the order of 10^6, such a large p means the worst case is extremely unlikely to occur in practice. With the population of the UK being approximately $6 * 10^7$ [**?**], the entire population would have to be scanned 10,000 times before a worst case scenario is likely to have happened.

Chapter 4

Project Management

Figure 4.1: The Hadley Scott Award For Creepiness.

Chapter 5

Testing

5.1 Person Isolation

Figure **??** shows the depth based cut off, described in Section **??**. As predicted, this cut off is sufficient to eliminate a large proportion of non-person data but does not remove the ring of points at the same depth as the person.

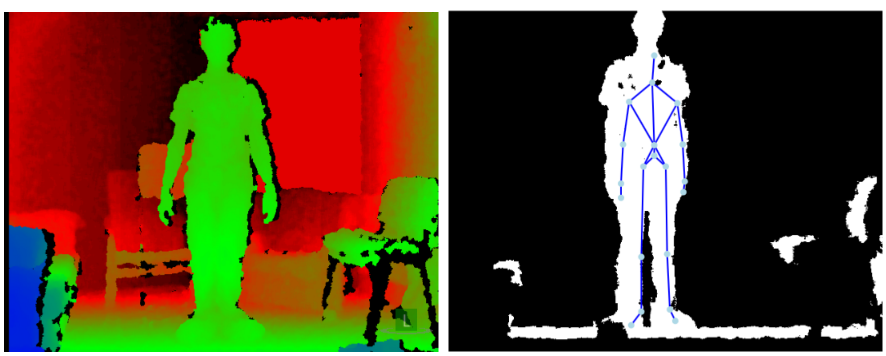

Figure 5.1: Depth Based Cut Off, before (left) and after (right).

Figure **??** shows the depth and hand based cut off, described in Section **??**. As predicted, the combination of these using depth and hand position is sufficient to isolate a person. The isolated person is coloured from black to white as depth increases away from the camera.

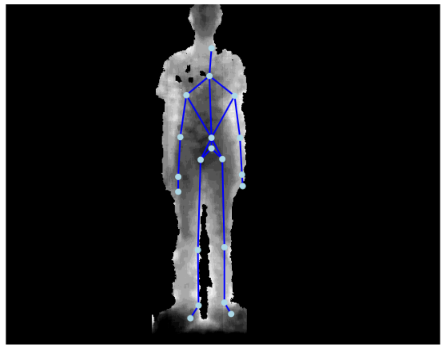

Figure 5.2: Depth and Hand Based Cut Off.

5.2 Volume Estimation

The development of Volume Estimation was split up into two phase of differing accuracy.

5.2.1 Minimum Bounding Box

The bounding box approach, described in **??**, was useful because it allowed the calculation of a transform between point cloud space (PCS) and real world space (RWS) without the added complexity of the volume rendering technique. This transform was necessary because the Kinect stores distances in terms of pixels rather than real world measurements such as meters and it is these real world measurements that the calculated volume should be outputted in. The transform is multiplicative and as such is of the form shown in Figure **??**.

$$Volume_{PCS} * TransformConstant = Volume_{RWS}$$

Figure 5.3: Transform between PCS and RWS

In order to facilitate the calculation of the transform constant, many people were scanned and their minimum bounding box calculated two ways. Firstly the box was calculated using the PARSE tool kit to give the box's volume in PCS. Secondly, the volume of the box was calculate in RWS by measuring the height, width and breadth of the person. These figures were then divided to determine the transform constant as in Figure **??**.

$$\frac{Volume_{RWS}}{Volume_{PCS}} = TransformConstant$$

Figure 5.4: Calculating the Transform Constant

Person	$Volume_{RWS}$ (m^3)	$Volume_{PCS}$ ($units^3$)	Transform Constant
Greg Corbett	0.23829	4.70440	0.05065
Bernard Sexton	0.16200	4.11551	0.03936

Figure 5.5: Table of Data Used to Calculate the Transform Constant

This approach is not without a drawback, the real world bounding box must be calculated by hand, using a tape measure. Such a process is inherently error prone, as highlighted in Section **??**. However it is hoped that, by taking as large a sample as possible and averaging the transform constants, such errors can be eliminated.

5.2.2 Volume Rendering

This section details the testing of the volume rendering approach, discussed in Section **??**

Chapter 6

User Manual