Abstract

Contrary to popular belief, skydiving is a competitive and technical sport, requiring careful control of body position in order to not only remain stable, but also to move around the sky in free-fall. A popular discipline in competitive skydiving is formation skydiving, where a group of skydivers form set shapes with their bodies whilst in free-fall. This Report proposes a software tool to be written; able to automatically judge formation skydiving footage. This will be done detecting the pose of each skydiver in the formation individually; this information will then be used to find the overall shape of the formation. This tool will combine and extend well established computer vision methods such as Active Shape Models, skeletonisation. The ultimate goal is to remove the need for manual competition judging, switching to a fully automated system.

1 Background

1.1 Formation Skydiving

Skydiving is a competitive and technical sport, requiring careful control of body position in order to move around the sky in a controlled manner whilst in free-fall.
A popular discipline in the sport is formation skydiving (FS). This involves multiple skydivers forming set shapes with their bodies, in order to score as many points as possible.

1.1.1 Rules

A point is awarded for each successful formation in a sequence ***(see figure 3)***. A successful formation is defined only by the correct hands of each skydiver holding the correct grips on the other skydivers ***(see figure 1)***, and not by the orientation of the skydivers themselves.

Figure 1: Diagram showing all of the available grips on a formation skydiving suit.

In 4 person FS (4-way FS) each skydiver has a specific slot in the formation. This is the position that they will always take when making each formation. This is done in such a way that the amount of movement required for each skydiver when transitioning between formations is minimised. The four slots in 4-way FS are known as Point, Tail, Outside Center (OC) and Inside Center (IC).

1.1.2 Training Practices

In order for the team to create formations in the sky, it is important that they practice the skydive multiple times on the ground. This is known as "dirt diving" and is often done using partially triangular wheeled platforms that each skydiver lays on, known as "creepers" ***(see figures 2a and 2b)***.

(a) FS practice platforms "creepers"

(b) "Creepers" in use while "dirt diving" a "Donut" formation

Figure 2: Images reproduced from `www.skydivespaceland.com/blog/images/creepers2.jpg` and `www.skydiveaz.com/images/old-images/creepers.jpg`

Another way that FS teams train is to use a vertical (indoor skydiving) wind tunnel. A fan is mounted at the bottom of the tunnel creating a strong wind that the skydivers can float on; simulating the environment of a regular skydive, but in a much more controlled environment ***(see figure 4)***.

1.2 Project Brief

The original focus of the project was to analyse video of dirt diving and identify which formations have been performed. The focus has been changed slightly, and is now to analyse footage of formation skydiving in a vertical wind tunnel. This change is largely due to the fact that the body position of a skydiver free-fall is much more similar to their body position in a wind tunnel, than when dirt diving. Another benefit of analysing footage taken in a wind tunnel is that the camera is mounted above the formation, remaining static ***(see figure 4)***. This allows the skydivers to much more easily be separated from the background via background subtraction[**?**]. The set of formations that will be analysed are the BPA rookie class of formations known as 'randoms'. The software must be able to recognise all 16 of the randoms ***(see figure 3)***.

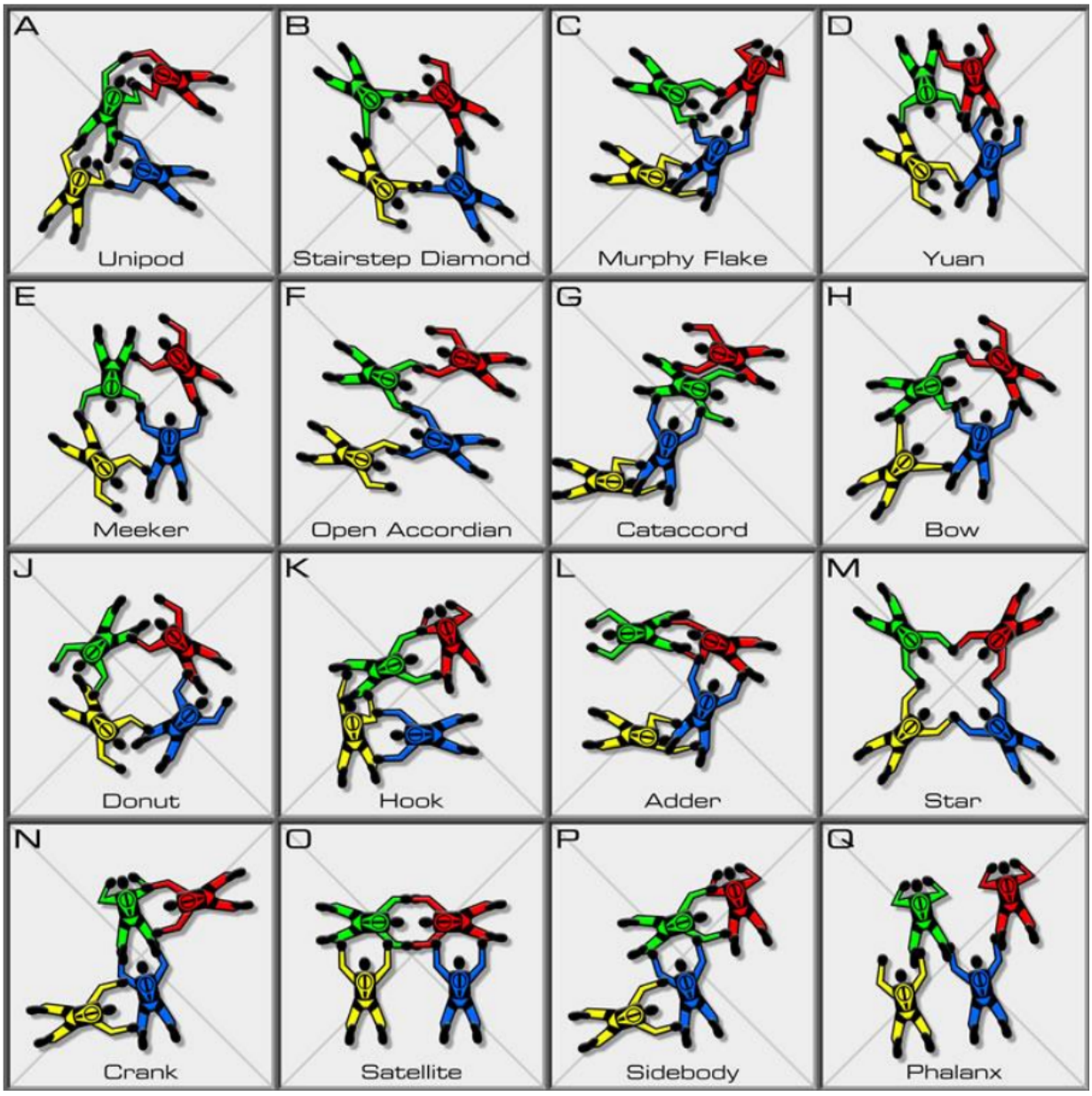


Figure 3: All 16 BPA rookie class FS 'randoms' formations, with their names. The colours for each slot are: Point red, OC green, IC blue, Tail yellow.
Reproduced from `http://www.teamsatori.co.uk/New%204W%20Random%20Dive%20pool.pdf`

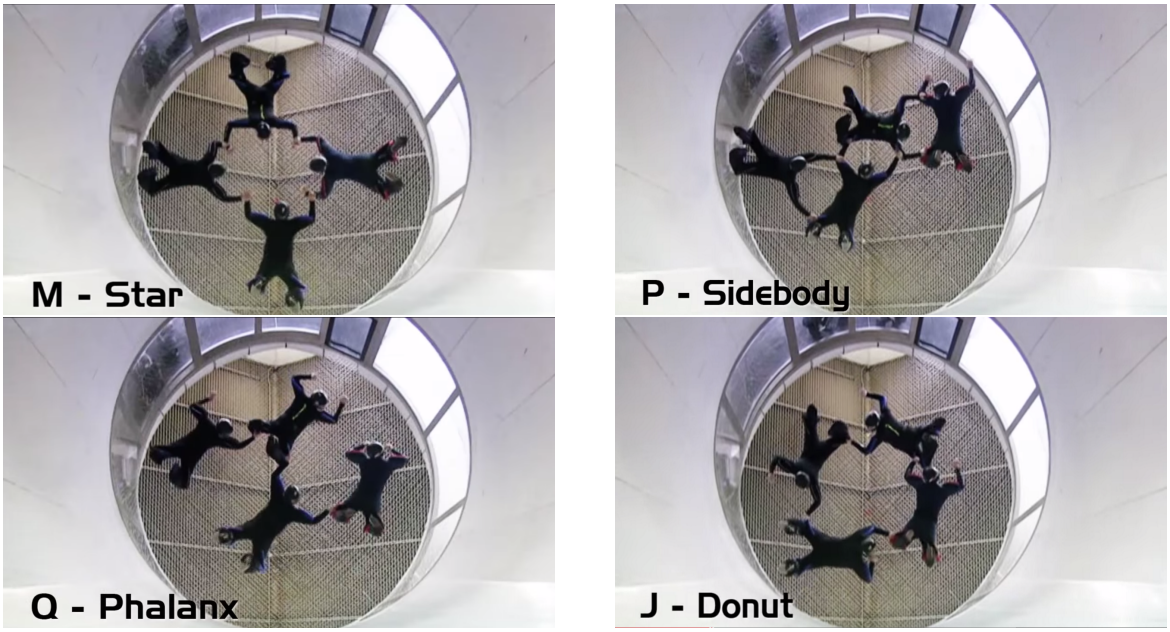


Figure 4: Sample FS formations performed in a wind tunnel, with their names.
Images reproduced from International Bodyflight Association, `www.youtube.com/watch?v=Y2B4S31Gf54&list=LLsEkKn0qzIHSGefmSGT_4og`.

An image of a skydiver in an FS body position can be simplified to a set of line segments between 11 key points; one for the waist, neck, head, left and right hands, elbows, knees and feet ***(see figure 5)***.

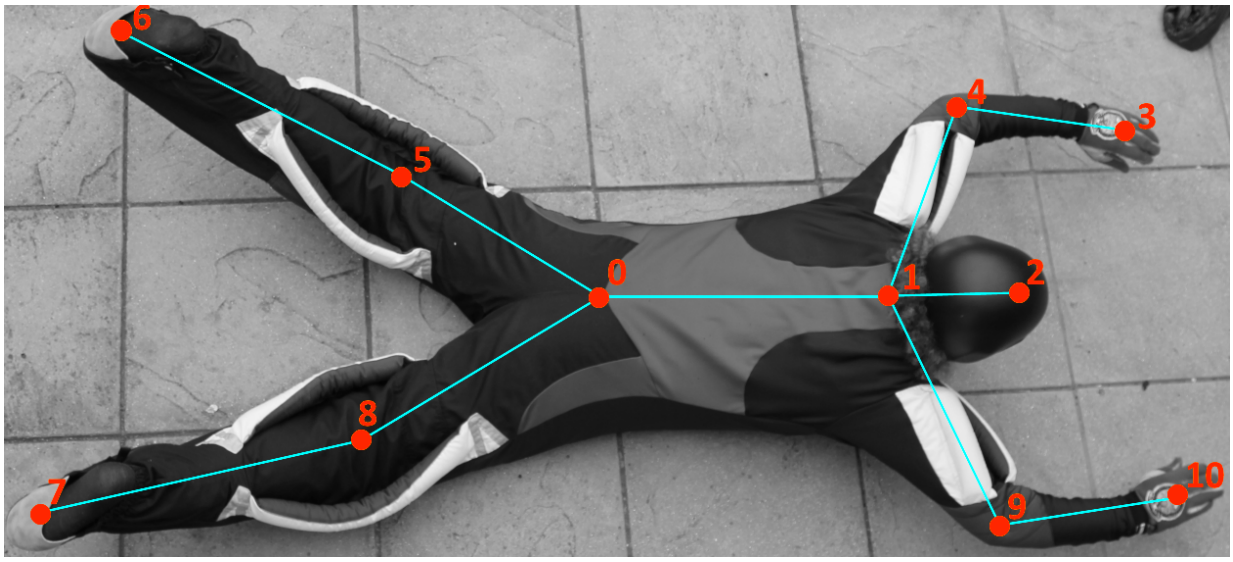


Figure 5: Diagram showing how an image of a skydiver can be simplified to a set of line segments between 11 points.

In order to complete this project C++ will be used, making use of the open source computer vision library OpenCV 3.0[] in order to reduce the time required to perform standard tasks, such as loading an image file, or receiving a video feed. If a graphical user interface (GUI) were to be implemented, the open source GUI library Qt 5.3[?] would be used. These libraries are platform independent; this should ensure that the program is not tied to any single operating system.

2 Literature Report

Due to the specialist nature of the project, relevant existing literature is somewhat limited. However, there does exist relevant literature on various methods for human posture recognition.

2.1 Posture Detection

Human pose detection can be a difficult problem to tackle; many factors contribute towards this, such as shadows, occlusions, complicated backgrounds and noisy source images.
There are many different proposed methods to tackle the task of human posture detection; over a wide variety of technologies. Some methods use depth cues extracted from 3d scanner information, such as [?] which uses a Microsoft Kinect to detect the human body. Other methods are designed to work with video[?], creating a 3d estimation of human pose in order to detect people. A common goal is to try to detect the pose of a person from a single static image. There are a variety of ways of doing this such as, using a model-based method, where an explicit pose model must be specified, often requiring complicated 3d modelling and rendering[?]. Another method used relies on texture cues in order to perform "Parts-based" object detection[?], this method neither uses an explicit model, or pre-labelled data. However, the method that is deemed to most fit goal of recognising the pose of a skydiver is an adaptive method, such as Active Shape Models. Active Shape Models are more often used in facial feature detection[?][?]; given that it only relies on a set of simple points, labelled from a set of training images it should be ideal for the task at hand.

2.2 Active Shape Models

Active Shape Models are statistical shape models that are able iteratively deform in order to fit a given example image. They can also be used to characterise the ways in which a set of points in an image can move. Using principal component analysis[?], it is possible to find a minimal set of feature variables which control the overall shape of an object. Varying these enables every possible permutation of the set of image points to be found.
In order for this to be done, a training set of points must be collected by hand from images similar to those that will be analysed. This This training set of points forms the initial point matrix.

$$M = \begin{bmatrix} x & y & x & y & \dots & \dots & x & y \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x & y & x & y & \dots & \dots & x & y \end{bmatrix}$$

Figure 6: Initial point matrix. Each row corresponds to a different image (blue). Pairs of columns contain coordinates of the same point across each of the images (red).

The iterative algorithm generalised procrustes analysis (GPA) is then used to optimally rotate, scale, and translate the points from each image. The goal of this is to have the points from each image match as closely as possible when superimposed on each other ***(see figure 7)***.

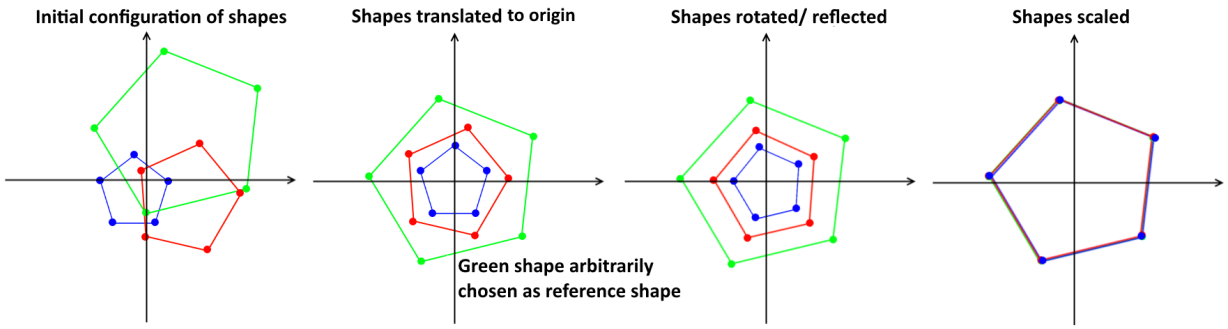


Figure 7: Performing generalised procrustes analysis on three shapes.

A shape matrix can be created using the same format as the initial point matrix, but containing the coordinates of the aligned image points. Finally, principal component analysis is used to reduce the dimensions of the shape matrix. This is done by finding the eigenvalues and eigenvectors of the covariance matrix. A more in depth explanation can be found at [**?**].

Using this method it is possible to find weather a skeleton of a skydiver is viable or not. This is done by creating the initial point matrix from a set of images of a skydiver in a variety of FS body positions. The resulting active shape model will contain information defining how a skydiver's body position can change. A set of test points that make up the skydiver's skeleton ***(see figure 5)*** can then be compared to this model to find if the skeleton is possible or not. This allows the skeleton to be recalculated, should it be found to be impossible.

3 Proposed Final Design

The proposed final design will take the form of a command line tool. It will take the path to a video of formation skydiving to be analysed as an argument. It's output will be a similar video file, but with an overlay showing which (if any) formations have been made at each stage in the video. It will also highlight the skeletons of the skydivers in different colours in order to show which slot in the formation they are flying. The skeletons will take the same colours as in ***figure 3***. Time permitting, a GUI may be implemented for the command line tool in order to make the software more user friendly.

Given that the project is entirely software based, additional support towards the project should not be necessary.

3.1 Flow Chart

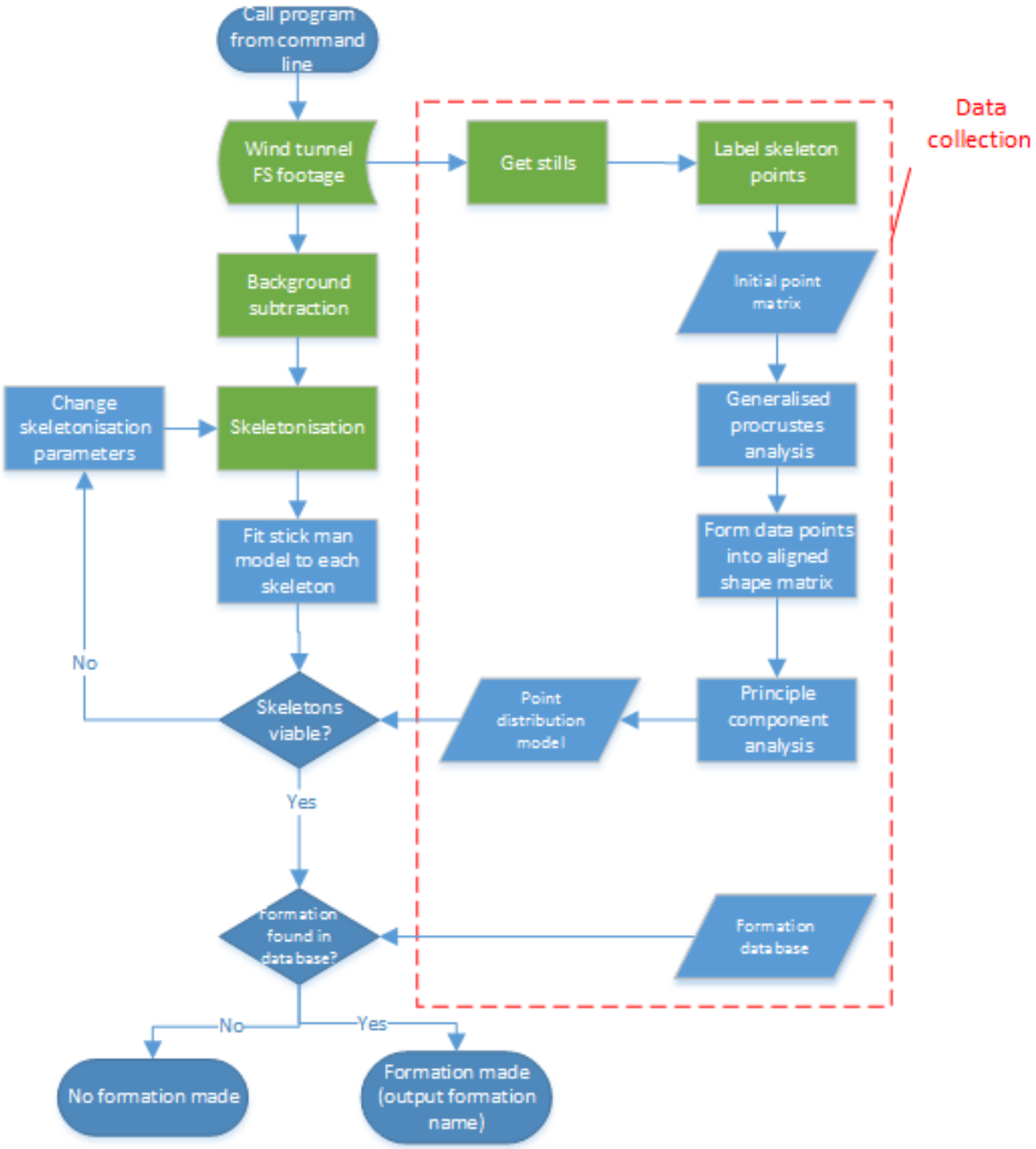


Figure 8: Flowchart showing overall functionality of proposed final design. Items shown in green have been completed. Items inside the red box are to be executed only once, and their results saved for later use.

4 Completed Work

4.1 Data Collection

Software has been written that handles collection of data for the initial point matrix to be used to create the point distribution model. In order to speed up data collection, the

program displays an image of a skydiver and asks the user to left click on different parts of the skydiver's body. The user can undo the last point by right clicking. Once all the points for the image are recorded, they are written to an output file and form the next row of the initial point matrix. Another image is then shown and the process repeats. This is a console based application and takes the file path of an image list as its only input ***(see figure 9)***. This image list is stored in an XML file, as OpenCV natively supports input and output to XML files. For each image, the program adds another row to the initial point matrix ***(see figure 6)***. Currently only 10 images have been used to create the initial point matrix, but this number will be increased before it is used to create the point distribution model.

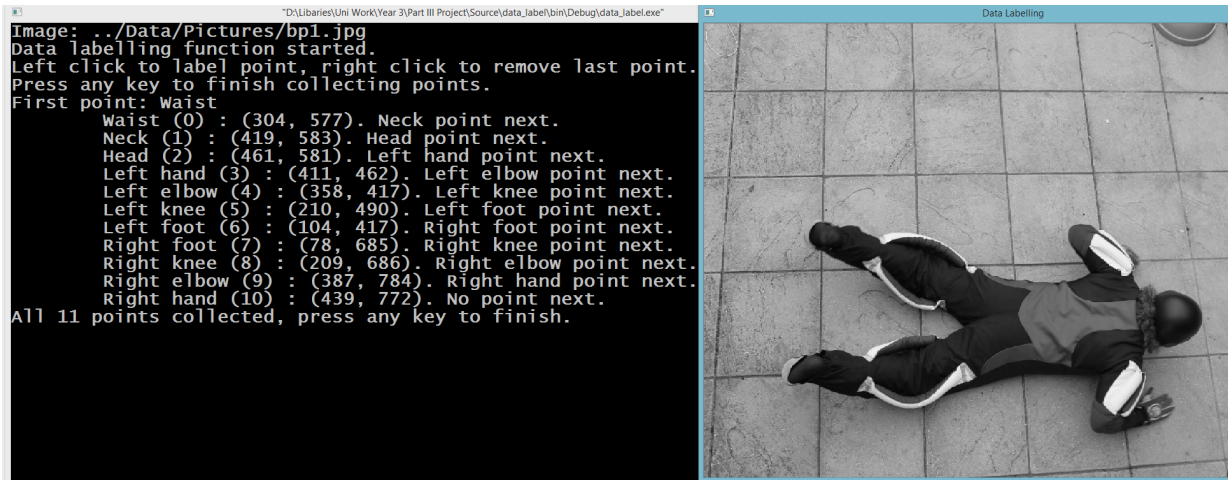


Figure 9: Data labelling program. Screenshot taken once all the required points of the image had been clicked on.

```
576,303,579,418,570,466,457,415,408,355,495,216,420,099,696,081,677,206,792,382,772,438
442,289,444,392,442,433,417,428,326,401,393,215,268,169,514,156,550,218,626,412,527,437
548,317,543,423,543,464,512,463,403,434,493,254,388,216,581,187,643,243,746,447,627,472
563,302,561,404,537,450,536,464,425,421,476,231,349,191,616,160,656,223,753,435,631,462
550,267,555,375,532,413,488,442,407,392,470,193,349,164,617,124,654,189,742,405,650,449
565,318,578,441,566,477,513,489,409,445,490,231,356,112,739,088,673,222,787,461,670,498
557,312,572,434,548,472,556,464,512,481,395,438,477,229,352,118,715,088,671,219,783,455
574,308,560,423,566,482,526,470,405,435,496,234,374,194,636,170,682,229,800,444,665,476
580,308,578,424,571,468,524,513,437,453,509,228,372,173,643,154,693,227,758,469,668,512
582,305,583,419,583,467,422,505,411,438,496,227,349,111,766,086,698,212,775,458,762,527
```

Figure 10: Matrix created when the program is supplied with a sample image list with 10 images in it.

4.2 Skeletonisation

Software has been written that creates a minimal, connected, 1 pixel thick skeleton from a binary mask. The function uses a medial axis transformation based method [**?**] to create the skeleton and the Ramer–Douglas–Peucker[**?**][**?**] (RDP) algorithm to minimise it.

4.2.1 Skeletonisation Algorithm

1. Find all boundary pixels (edge of the binary mask) using a binary laplacian mask[**?**]. For each boundary pixel do the following:

2. Test pixels neighbours

p8	p1	p2	y
p7	p0	p3	
p6	p5	p4	↓
x		→	

3. Count Np0, number of non-zero neighbours of p0.

4. Count Tp0, number of 0-1 transitions in sequence p1, p2, ..., p8.

5. Check the initial conditions; mark for deletion any point that satisfies them all.
   Initial conditions:

   - $cA : 2 <= Np0 <= 6$
   - $cB : Tp1 = 1$
   - $cC : p1.p3.p5 = 0$
   - $cD : p3.p5.p7 = 0$

6. Delete any points marked for deletion.

7. For each remaining boundary pixel, recalculate Np0 and Tp0, then check secondary conditions; marking for deletion any point that satisfies them all:
   Secondary conditions:

   - $cA : 2 <= Np0 <= 6$
   - $cB : Tp1 = 1$
   - $cC_{-} : p1.p3.p7 = 0$
   - $cD_{-} : p1.p5.p7 = 0$

8. Delete any points marked for deletion.

Repeat process until no points are deleted.
For more a more in depth explanation of the algorithm, see [**?**].

Initially a simpler algorithm[**?**] was used in order to create the skeleton using morphological operations to perform thinning. However, this algorithm did not ensure that the skeleton created was minimal or connected, and was therefore insufficient.

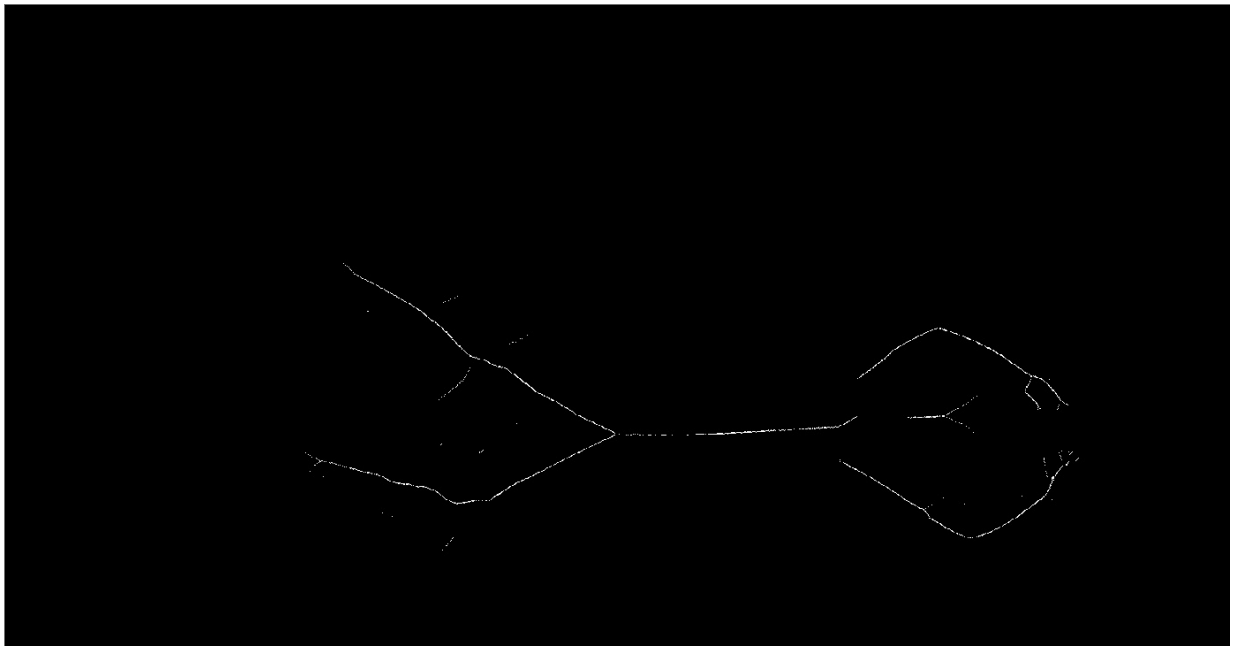


Figure 11: Result of using the morphological thinning algorithm on the binary image shown in ***figure 13b***.

4.2.2 Skeleton Simplification Algorithm

The RDP algorithm is designed to simplify a curve, while still maintaining its overall shape. A curve can be thought of as a collection of linked line segments, with each line segment stretching between two consecutive points on the curve. The basic principle of the algorithm is to reduce the number of line segments in a curve; creating a similar, but simplified curve. The algorithm is initially given an ordered array of all the points in the curve. The algorithm is also given the variable ϵ; this defines the degree to which the curve should be simplified and represents a minimum distance between line segments on the curve. It marks the start and end points of the curve as the first line segment (L), and then finds the point furthest from this initial line segment. If the distance from this point to the line segment (d) is less than ϵ then it is marked for removal. If the point is further than ϵ then it is kept and the algorithm recursively called with this point as the new end point. In this way the algorithm is able to remove every point that is closer than ϵ to the line segment that joins its neighbours.

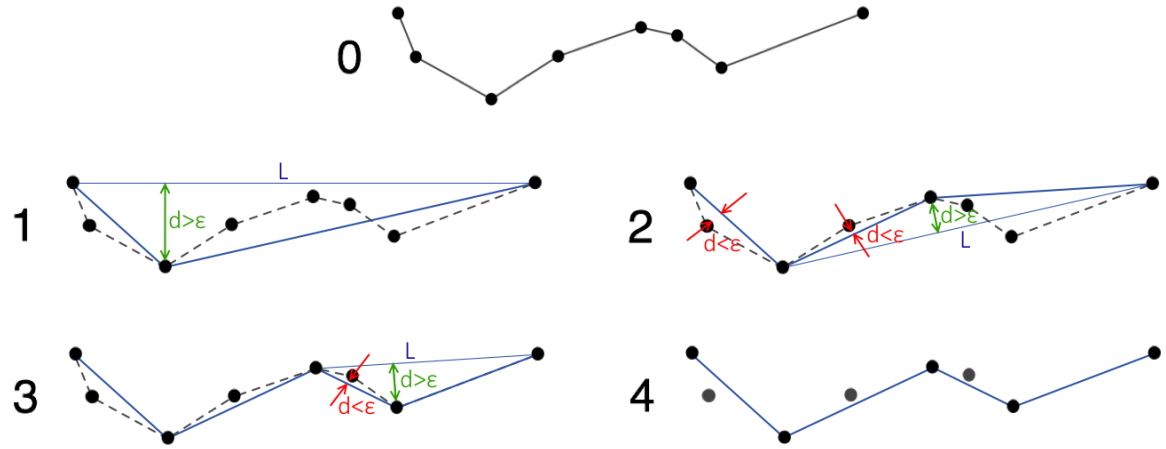


Figure 12: Diagram showing the RDP algorithm working on a curve. 0 being the initial curve to be simplified, and 4 being the fourth result the algorithm. Red arrows show points marked for deletion, green arrows show points to be kept
Modified from `http://upload.wikimedia.org/wikipedia/commons/9/91/Douglas_Peucker.png`.

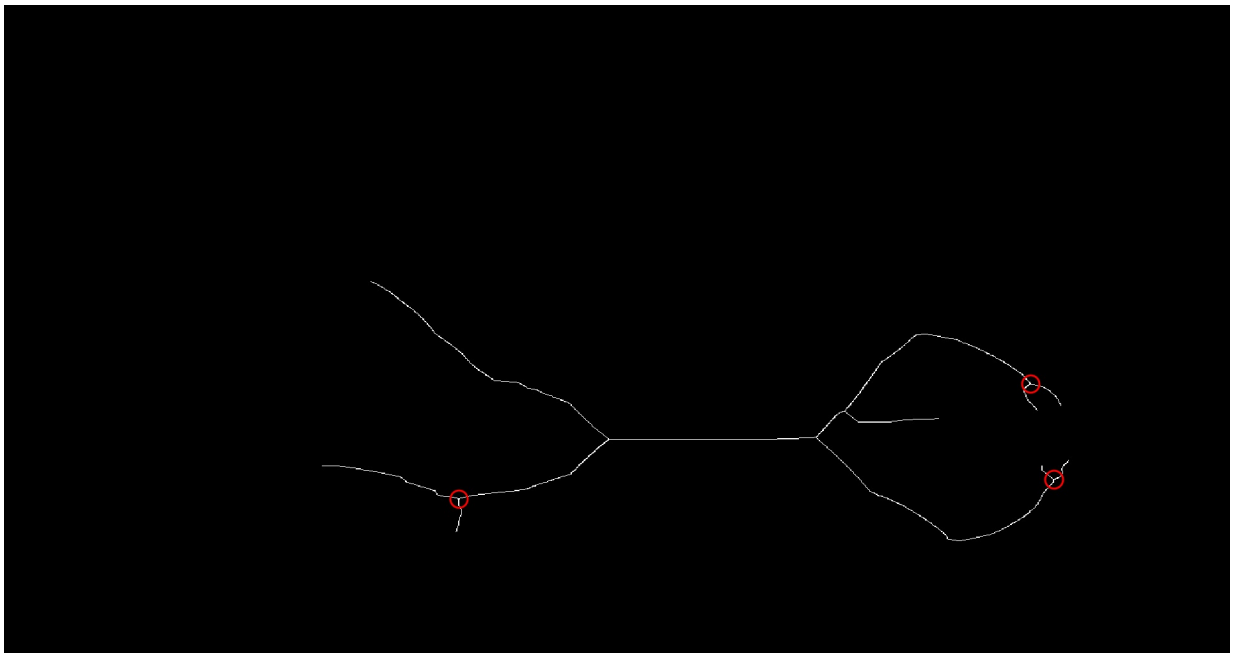
The RDP algorithm is used to reduce the number of points in the skeleton, simplifying it; however, it is only able to simplify curves with no bifurcations. This proves to be a problem when trying to simplify a skeleton produced by the skeltonisation algorithm, as it will always have bifurcations at the waist and neck. It may also also produce erroneous bifurcations at other points on the skeleton. These bifurcations can be seen in ***figure 13c***. The location of these bifurcations can be detected by applying a modified version of condition cB from the skeletonisation algorithm. Tp0 is calculated in the same way, by looking at the number of 0-1 transitions in the sequence p1, p2, ... , p8.

p8	p1	p2	y
p7	p0	p3	
p6	p5	p4	↓
x		→	

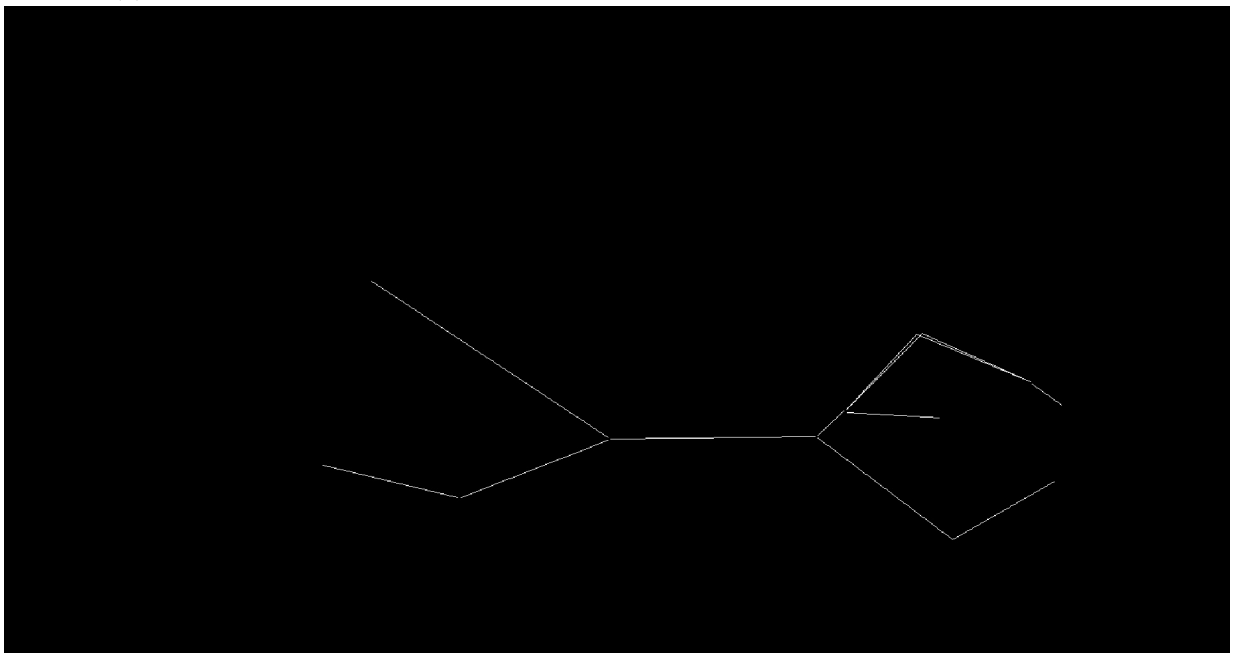
If $Tp0 > 2$ then a bifurcation is present at p0. This is always true, as the skeletonisation algorithm ensures that the skeleton is a single pixel thick 8-connected component.
Once the location of every bifurcation is found, the skeleton can be split into separate curves, by removing the bifurcation point. The RDP algorithm is then run on each of these curves independently. The simplified curves can then be recombined to form a simplified skeleton. At this point it is also possible the remove the false skeleton sections created by the skeletonisation algorithm forming erroneous bifurcations. The simplest way to do this is to delete any simplified skeleton section that is shorter than a given length. This works as the false skeleton sections tend to be shorter than the correct ones. A more reliable method for detecting false skeleton sections could be used in the final design.

(a) Skeletonisation source image.

(b) Binary mask of source image.



(c) Result of skeletonisation function. Erroneous bifurcations are circled in red.



(d) Result of skeleton simplification algorithm.

Figure 13: Results of skeletonisation and simplification algorithms.

4.2.3 Background Subtraction

A background subtraction algorithm has been implemented using the OpenCV libraries[**?**]. However, this cannot be properly tested, as the stock wind tunnel footage currently being used is too noisy.

5 Project Management

In order to properly manage the project, weekly meetings are conducted between the student and supervisor; progress and possible next steps are discussed.
An initial Gantt chart was created in order to aid in time management ***(see appendix A)***. However, as the project progressed this Gantt chart was seen to be irrelevant and A new chart was produced ***(see appendix B)***.

5.1 Risk Management

A risk assessment was carried out in order to assess potential problems, and how they could be prevented/ resolved.

Potential Problem	**Loss (1-5)**	**Probibility (1-5)**	**Risk**	**Plan**
Unable to get suitable stock FS footage	4	3	12	Contact FS coaches to ask for stock footage. If this is not possible, record a training session in the wind tunnel with my 4-way FS team.
Source code lost due to home computer failure	5	1	5	All source code and project documentation is backed up on a private github repository.
Severe Illness	4	1	4	Time is factored into the initial Gantt chart in order to account for unforeseen problems, such as illness.
Creating an active shape model is more difficult and time consuming than expected	4	3	12	More time can be contributed to the project, should this be the case.
Fitting a stick man model to a skeleton is more difficult and time consuming than expected	4	3	12	More time can be contributed to the project, should this be the case.
No project work is completed during the exam period, and the project falls behind schedule	3	4	12	After the exam period, extra time can be contributed to the project.