Electronics and Computer Science
Faculty of Physical and Applied Sciences
University of Southampton

Merlin Webster
December 8, 2014

Skydiving Formation Recognition

Project supervisor: Dr Jonathon S Hare
Second examiner: Prof Lie-Liang Yang

A project progress report submitted for the award of
MEng Electronic Engineering

1 Introduction

2 Background

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 freefall in a controlled manner.
A popular discipline in the sport is formation skydiving (FS). This involves multiple skydivers forming set formations in freefall, in order to score as many points as possible. A point is awarded for each successful formation in a sequence ***(see figure ??)***. 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 in order to make each formation. This is done in such a way to minimise the amount of movement required for each skydiver when transitioning between formations. The four slots in 4-way FS are Point, Tail, Outside Center (OC) and Inside Center (IC). 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 ?? and ??)***.

(a) FS practice platforms "creepers"

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

Figure 1: Images reproduced from `www.skydivespaceland.com/blog/images/creepers2.jpg` and `www.skydiveaz.com/images/old-images/creepers.jpg`

2.1 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 modified, and is now to analyse footage of formation skydiving in a vertical wind tunnel, indoor skydiving. This change is due to the fact that the body position of a skydiver in the wind tunnel is exactly the same as that of a skydiver in freefall. The wind is turned up to freefall speeds, simulating the environment of a regular skydive, but in a much more controlled environment. Another benefit of analysing footage taken in a wind tunnel is that the camera is mounted above the formation, and remains static ***(see figure ??)***. The set of formations I will be analysing are the BPA rookie class of formations known as 'randoms'. The software must be able to recognise all 16 of the randoms ***(see figure ??)***.

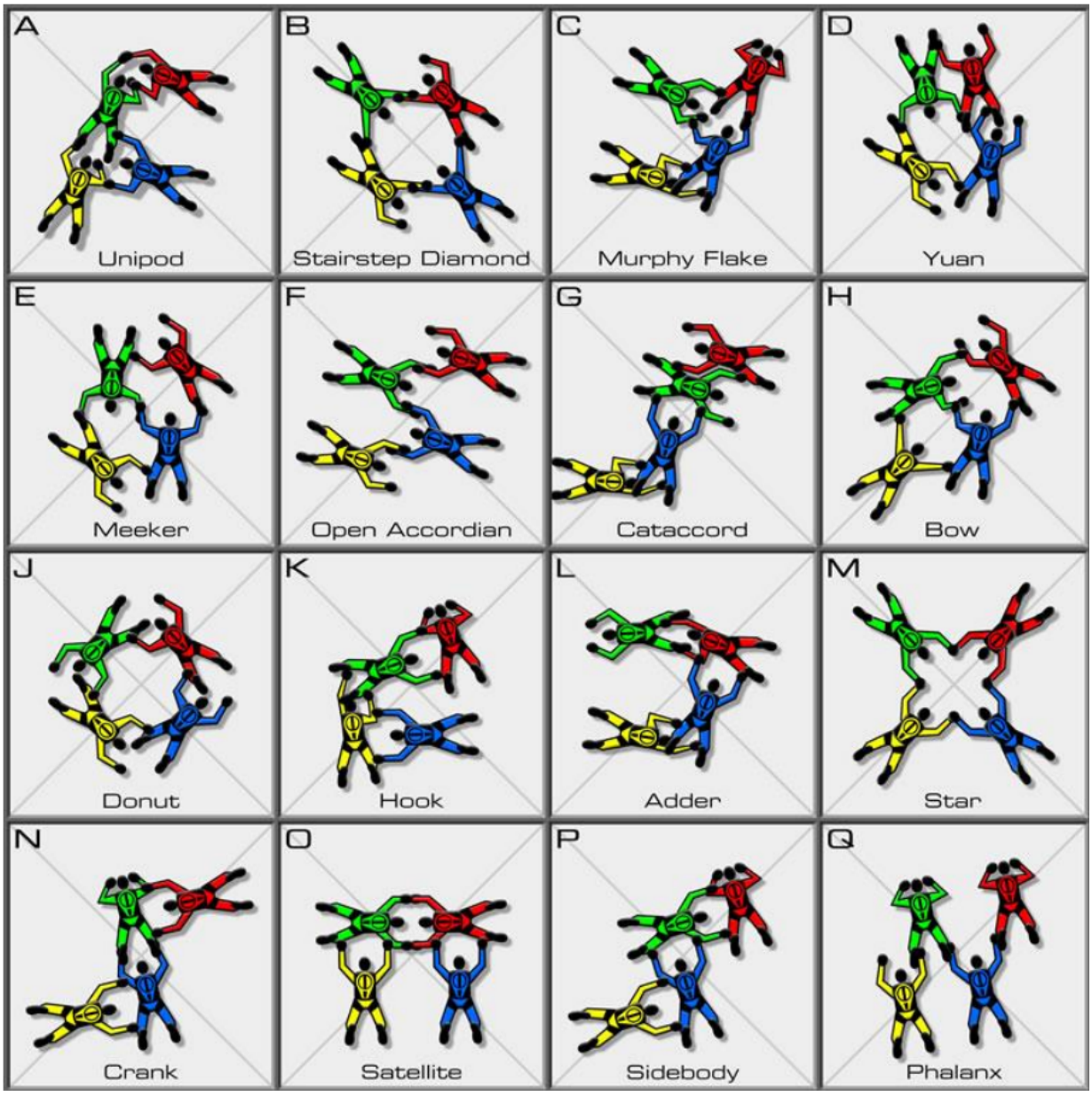


Figure 2: All 16 BPA rookie class FS 'randoms' formations, with their names. The colours for each slot are: Point red, OC green, IC blue, Tail yellow.
Reproduced from `http://www.teamsatori.co.uk/New%204W%20Random%20Dive%20pool.pdf`

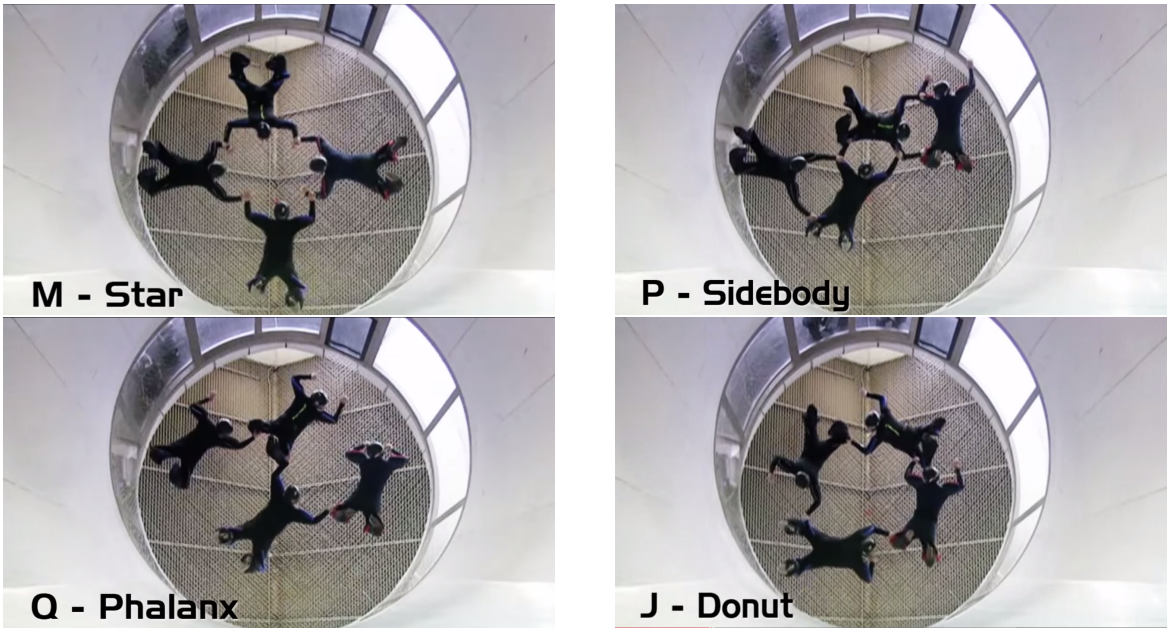


Figure 3: Sample FS formations performed in a wind tunnel, with their names.
Images reproduced from International Bodyflight Association, `www.youtube.com/watch?v=Y2B4S31Gf54&list=LLsEkKn0qzIHSGefmSGT_4og`.

3 Literature Report

List sources... How does generalised procrasts analysis work?
How do active shape and point distribution models work?
What am I using for skeletonisation?

4 Proposed Final Design

The proposed final design will take the form of a command line tool. It will take a video of formation skydiving to be analysed. It will output the same video file, but with an overlay showing which formations have been formed. 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 **??**.

5 Completed Work

5.1 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 skydivers 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 ??)***. 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. This contains the coordinates of each point x1, y1, x2, y2, ... ,x11 ,y11 and stores the data for each new image in another row. 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.

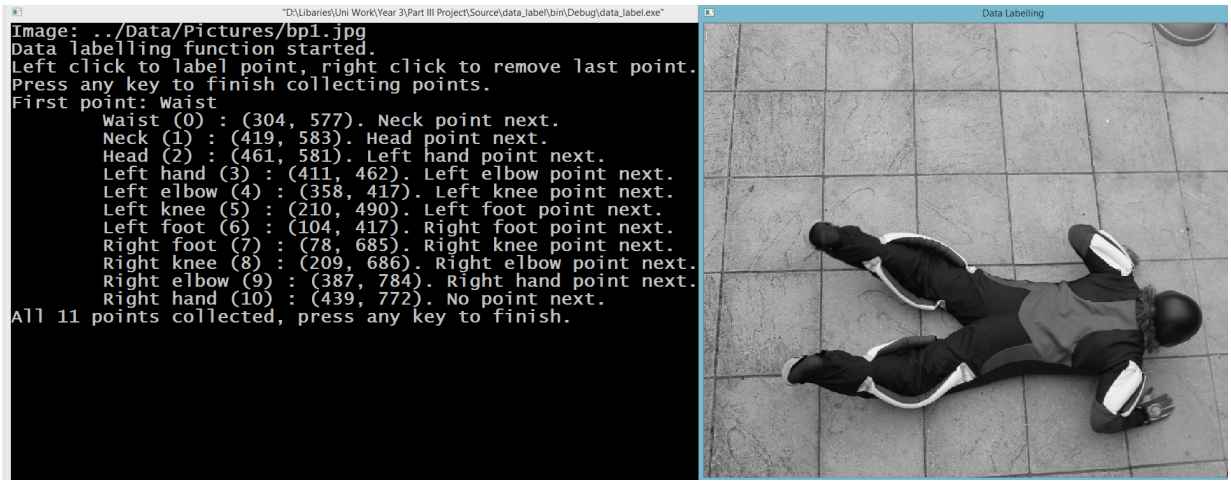


Figure 4: Data labelling program. Screenshot taken once all the required points of the image had been clicked on.

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

5.2 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.

5.2.1 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 [**?**].

5.2.2 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.

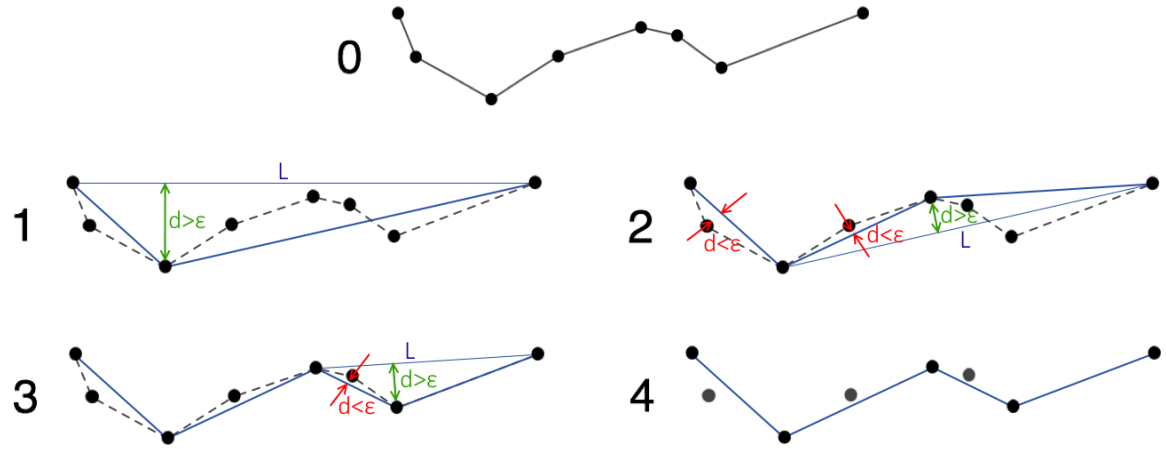


Figure 6: Diagram showing the RDP algorithm working on a curve. 0 being the initial curve to be simplified, and 4 being the fourth result the algorithm. Red arrows show points marked for deletion, green arrows show points to be kept
Modified from `http://upload.wikimedia.org/wikipedia/commons/9/91/Douglas_Peucker.png`.

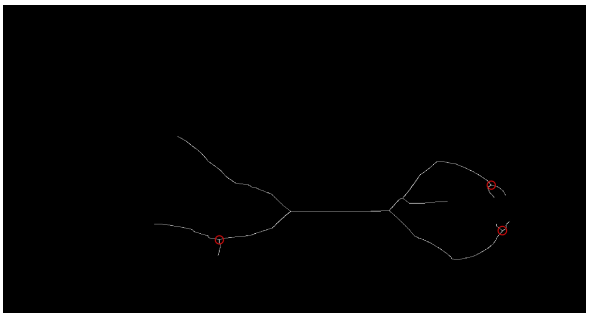
The RDP algorithm is used to reduce the number of points in the skeleton, simplifying it. However, it is only able to simplify curves with no bifurcations. This proves to be a problem when trying to simplify a skeleton produced by the skeltonisation algorithm, as it will always have bifurcations at the waist and neck. It may also also produce erroneous bifurcations at other points on the skeleton. These bifurcations can be seen in ***figure ??***. 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		→	

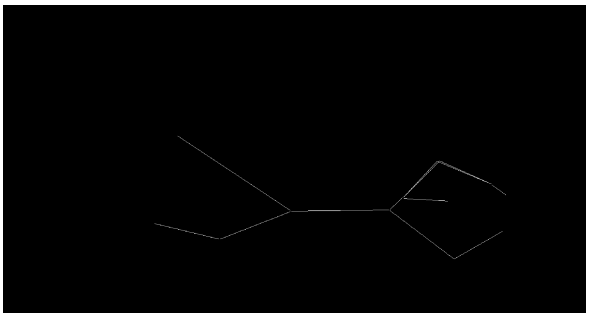
If $Tp0 > 2$ then a bifurcation is present at p0. This is always true, as the skeletonisation algorithm ensures that the skeleton is a single pixel thick 8-connected component.
Once the location of every bifurcation is found, the skeleton can be split into separate curves, by removing the bifurcation 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 7

6 Plan of Remaining Work

An enhanced project description (development of the brief above).
A report on the background research and literature search.
The proposed final design of the system or experiment.
A justification for the approach.
An account of the work to date.
A plan of the remaining work.
An estimate of any support required to complete the project.
A Gantt chart showing the schedule of both completed and remaining work.

References

[1] Rafael C. Gonzalez & Richard E. Woods, "Skeletons", Digital Image Processing – Second Edition — International Edition (2001), pg650-653

[2] Rafael C. Gonzalez & Richard E. Woods, "The Laplacian", Digital Image Processing – Second Edition — International Edition (2001), pg581-585

[3] Urs Ramer, "An iterative procedure for the polygonal approximation of plane curves", Computer Graphics and Image Processing (1972), pg 244–256

[4] David Douglas & Thomas Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", The Canadian Cartographer (1973), pg112–122

Appendices

Gantt Charts

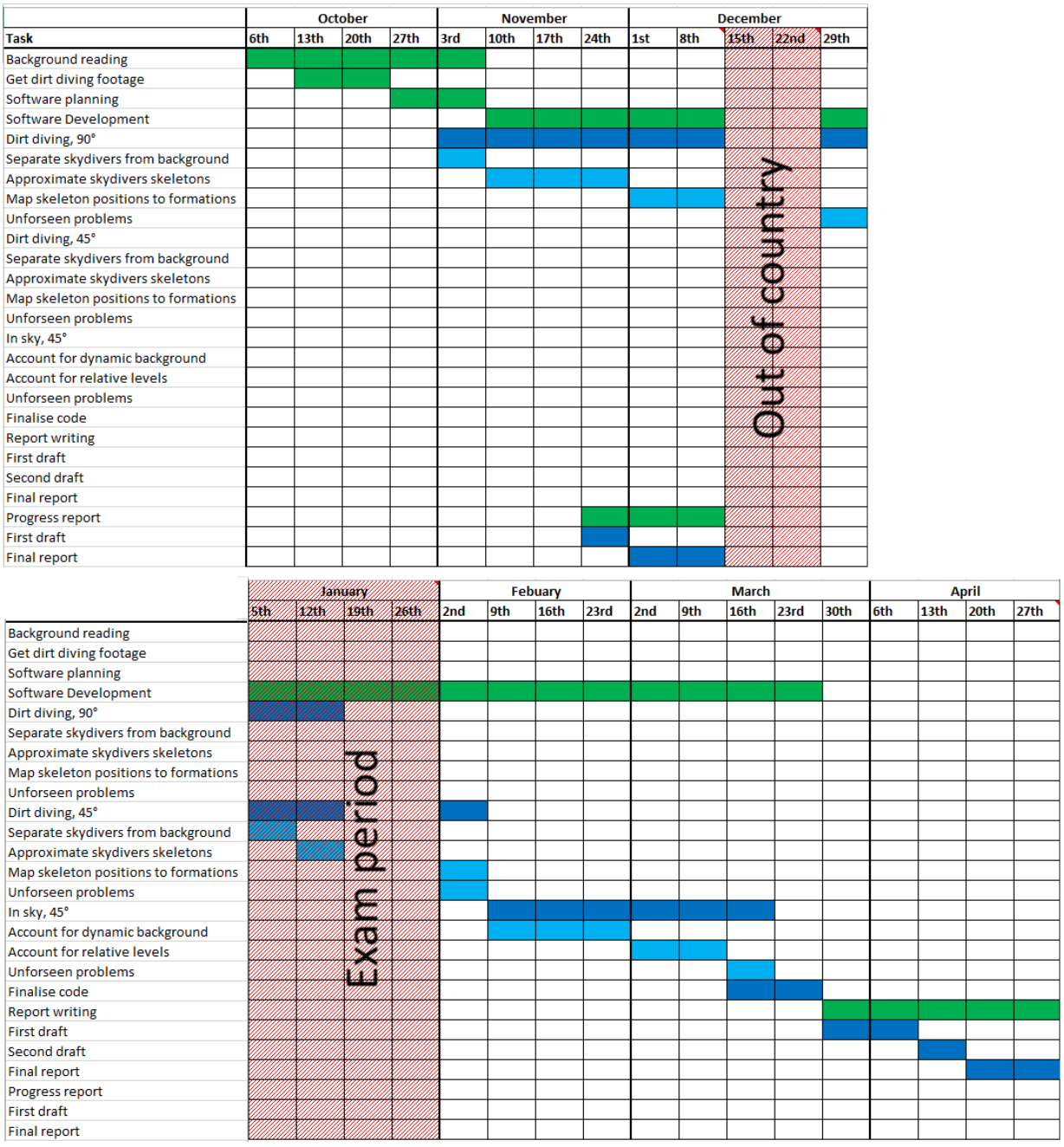


Figure 8: Initial Gantt Chart