Video Capture for a Tethered Balloon

by

Richard Shaw (PEM)

Fourth year undergraduate project in
Group C, 2011/2012

I hereby declare that, except where specifically indicated, the work submitted is my own original work.

Signed:

Date:

Abstract

This project explores the use of multiple cameras to measure optically the low frequency vibration of a tethered balloon. The report presents a low-cost experimental method for capturing the dynamic behaviour of the balloon-tether to validate existing numerical models.

The context for this study is the SPICE project – Stratospheric Particle Injection for Climate Engineering. SPICE investigates the feasibility of solar radiation management through the deployment of reflective aerosol particles into the stratosphere. Particles are proposed to be delivered to the stratosphere through a high-pressure pipe suspended by a balloon tethered at an altitude of 20km. The camera based system described in this paper is designed to monitor the dynamics of an initial test – a tethered balloon at a height of 1km.

Measurements are to be obtained using a series of commercially available cameras with telescopic lenses positioned at fixed points far from the base of the tether. These are used to record the dynamics of the tether as it moves in the wind, either as video or a sequence of still photographs to achieve higher resolution. The main difficulty arises from the fact that the tether is 1km long and only 13mm in diameter and therefore will be barely visible in an image. To get around this, it is proposed that multiple cameras can be used to cover the full length of the tether.

Various image processing techniques are discussed to extract the pixels corresponding to the position of the tether from the image sequence/video on a frame-by-frame basis. The techniques are based around the principle of searching for the darkest pixels, and include thresholding and filtering to improve the signal-to-noise ratio. The effectiveness and accuracy of these techniques is discussed and evaluated.

Using multiple cameras in a stereo set-up enables the 3D coordinates of discrete points along the length of the tether to be determined for each frame. Synchronisation and calibration of multiple cameras is discussed. The ability to track the motion of specific points on the tether over a number of frames is explored. Hence the displacement, velocity and acceleration of points along the tether can be found from the video/image sequence at frame rates respecting the Nyquist frequency.

Small-scale tests were carried out in the lab using hanging cables and small tethered helium balloons. The 2D motion of a hanging chain was observed using a single camera and its position extracted over a number of frames. The displacement-time graphs of several

points along the chain are plotted and the frequency of oscillation agrees fairly well with the theoretical value. Using two cameras in stereo, the 3D position of a hanging cable was found.

Larger-scale outdoor tests using kites were performed in an effort to represent the conditions of the actual full-scale tethered balloon, which would have a sky background. The kite string was able to be extracted successfully from the images using the filtering method. With two cameras, the 3D position of the points along the string could be found for each frame.

Although significant progress has been made, there is still work to be completed on the system. Currently, the system is not able to identify a specific point on the tether and track its position in 3D from one frame to the next. To do this, it may be possible to integrate from a fixed point to a known length along the tether, under the assumption that the tether is inextensible. Alternatively, white or reflective markers along the length of the tether may have to be used. Work also needs to be done to improve the robustness and efficiency of the tether detection software.

Table of Contents

1. Introduction

In a recent report, the Royal Society published the findings of a major study into geoengineering the climate[1], in which it states that the impacts and costs of man-made climate change will be "large, serious and unevenly spread" and that these impacts can be "reduced by adaptation and moderated by mitigation, especially by reducing emissions of greenhouse gasses".

The president of the Royal Society, Martin Rees has said, "The continuing rise in the atmospheric concentration of greenhouse gases, mainly caused by the burning of fossil fuels, is driving changes in the Earth's climate. The long-term consequences will be exceedingly threatening, especially if the nations continue at the current rate in the coming decades. Most nations now recognise the need to shift to a low-carbon economy, and nothing should divert us from the main priority of reducing greenhouse gas emissions. But if such reductions achieve too little, too late, there will surely pressure to consider a 'plan B' – to seek ways to counteract the climate effects of greenhouse gas emissions by geoengineering"[2].

From doing this project, it has become apparent that there seems to be a great deal of confusion and doubt amongst both scientists and the general public over the subject of geoengineering. And while there has been some resistance against progressing with geoengineering, there remains a significant interest in possible geoengineering solutions with many different techniques being proposed; some being too far-fetched or potentially dangerous, while others are more credible and are currently being researched. The Royal Society report[1] says that geoengineering can be divided into two main categories:

1. Carbon Dioxide Removal (CDR) techniques, which remove CO2 from the atmosphere. As they address the root of climate change, rising CO2 concentrations, they have relatively low uncertainties and risk. However, these work slowly to reduce global temperatures.
2. Solar Radiation Management (SRM) techniques, which reflect a small percentage of the sun's light and heat back into space. These methods act quickly, and so may represent the only way to lower global temperatures quickly in the event of a climate crisis. However, they only reduce some, but not all, effects of climate change, while possibly creating other problems. They also do not affect CO2 levels and therefore fail to address the wider effects of rising CO2, including ocean acidification.

The second of these categories is where the SPICE project comes in – Stratospheric Particle Injection for Climate Engineering. SPICE investigates the feasibility of solar radiation management through the deployment of reflective aerosol particles into the stratosphere which reflects the sun's rays, essentially mimicking the global cooling effect of a volcanic eruption. Studies based on the evidence from volcanic eruptions have shown that the sulphate particles released into the stratosphere result in reduced globally averaged surface temperatures. The sulphate particles are proposed to be delivered to the stratosphere through a high-pressure pipe suspended by a balloon tethered at an altitude of 20km, as shown in Fig. 1.

The tethered-balloon delivery system would represent the tallest man-made structure in history[3], although the concept of a tethered-balloon is not a particularly radical technology and there are many examples of tethered-balloons being used in the past and today, including barrage balloons, tethered helium passenger balloons, and weather balloons. The tether will be around 25km long, in the form of a reinforced pipe able to withstand extreme weather and high pressure loadings.

A 1km test-bed, including scaled-down versions of the balloon, tether and pumping system, is to be launched to validate existing computer simulations of the balloon and tether dynamics as it moves in the wind. The test-bed will consist of a 15m diameter balloon, as shown in Fig. 2, and a 1km length of pipe. The system will be used to spray only water droplets into the atmosphere rather than actual sulphate particles, i.e. it is purely a test of balloon and tether dynamics, not geoengineering.

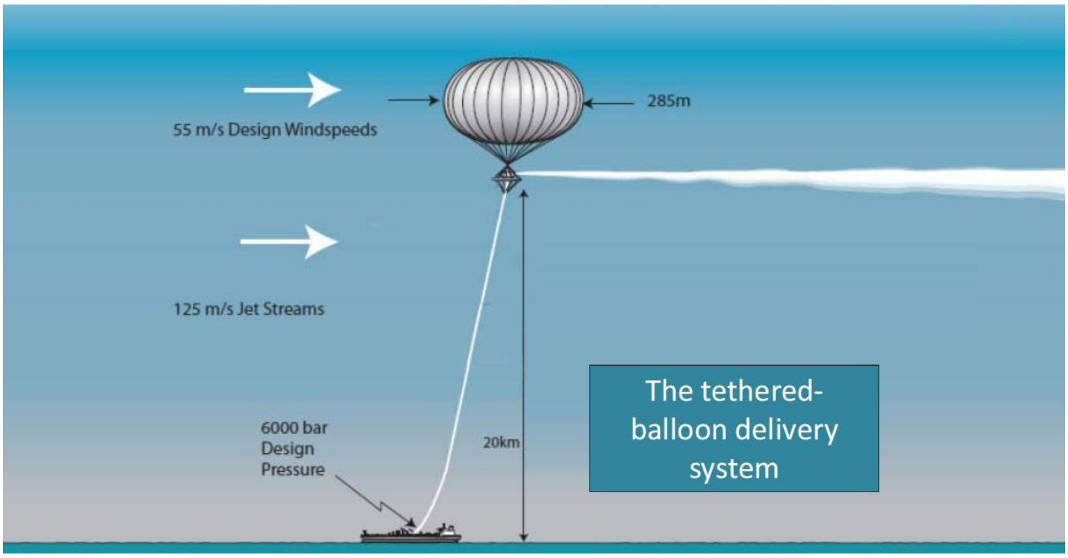

Figure 1. Tethered-balloon delivery system.

An experimental method of measuring and analysing the tether dynamics is therefore required. Various methods of doing this have been considered, including using accelerometers attached along the 1km length of tether to record the acceleration, and hence the velocity and displacement of discrete points along the tether. However, this would be a difficult and time consuming process as the accelerometers would have to be manually fixed to the tether, a challenge in itself when dealing with 1km of pipe. The tether would have to be modified to accommodate the accelerometers, with wires possibly running through the inside of the tether. Accelerometers would also affect the aerodynamics of the tether if fixed on its exterior.

The aim was to devise a simpler system which would leave the tether completely unaffected, so as not to alter its dynamics and for reasons of convenience.

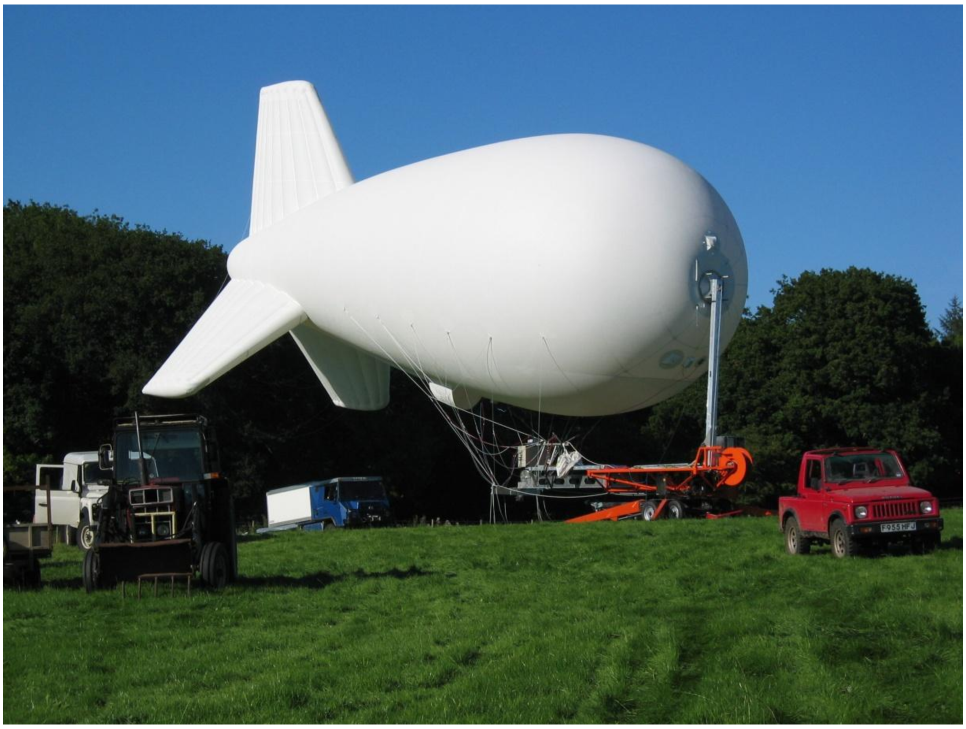

Figure 2. 15 metre diameter balloon test-bed.

It was proposed that the most effective way of monitoring the tether dynamics, without affecting the tether itself, would be to use a camera-based system to measure the low-frequency vibration of the tether optically. There have been many instances of using camera technology to measure vibration, for example for monitoring the vibration in the spans of suspension bridges[4].

The idea is to use multiple cameras equipped with telephoto lenses, positioned at some fixed point on the ground away from the base of the tether, as shown schematically in Fig. 3. The cameras would be used to capture the dynamics of the tether, in conjunction with some specially designed software to extract the position, velocity, and acceleration data from the acquired footage of the tether. The extracted data could then be compared to the existing computer simulations of the tether to validate the model. If the model fits the experimental data well, there would be greater confidence in the accuracy of the model which will help predictions to be made about larger-scale designs.

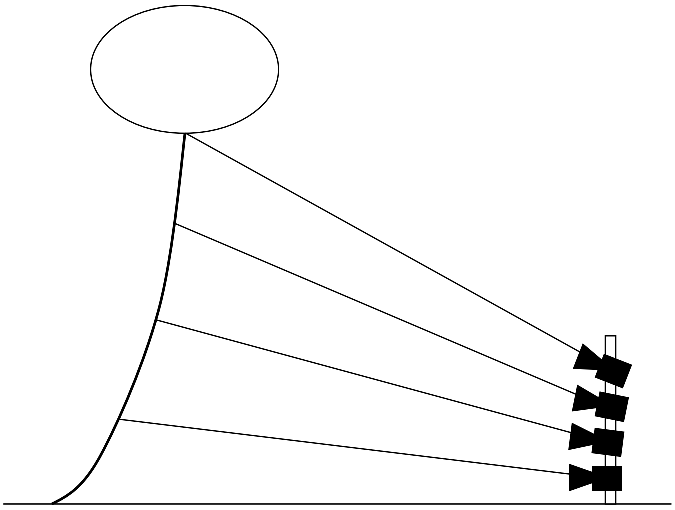

Figure 3. Camera-based system schematic.

Unfortunately the full-scale launch has been delayed indefinitely, but the system has still been designed with the application of the full 1km launch in mind. Meanwhile, the designed system has been tested experimentally using strings, small-scale tethered helium balloons and kites.

2. Methodology

This section of the report describes the methodology and theory behind designing the camera-based system for monitoring the dynamics of the balloon tether, and some of the challenges this involves. The proposed system is described in detail; describing the image processing theory for the designing the algorithm to extract the tether position and the use of common computer-vision techniques using stereo cameras to recover the motion in 3D.

Tether detection

A significant challenge of using a camera-based system to record the dynamics of the balloon-tether is the detection and extraction of the tether itself from video or a sequence of images, in order to track its motion over time. The tether is very long and narrow: it measures 1km in length and is only 13mm in diameter. It will therefore be difficult to detect in an image as the resolution of the camera lens will be the limiting factor. Assuming a single 1 megapixel camera (a sensible assumption for video) with a square aspect ratio (1000px by 1000px), is used to observe the entire 1km length of tether, as shown in Fig. 4, then 1 pixel is equivalent to a 1m length of tether. Since the tether is 13mm in diameter, and assuming square pixels, it will subtend approximately 1/100th of a pixel. Clearly a camera with this resolution will not be sufficient to detect the tether.

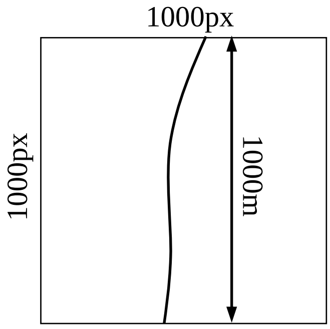

Figure 4. Image frame schematic.

However, since no camera lens is perfectly sharp, the tether will actually subtend a greater pixel width than this. A lens will tend to blur the pixel intensity of an image slightly in a Gaussian way. For example, an object which is equivalent to 1 pixel in size may be spread over several pixels, with the pixel intensity outwardly decreasing from the centre. So for the tether, which might be sub-pixel in size, may actually be spread over a few pixels but with lower intensity, as shown in Fig. 5.

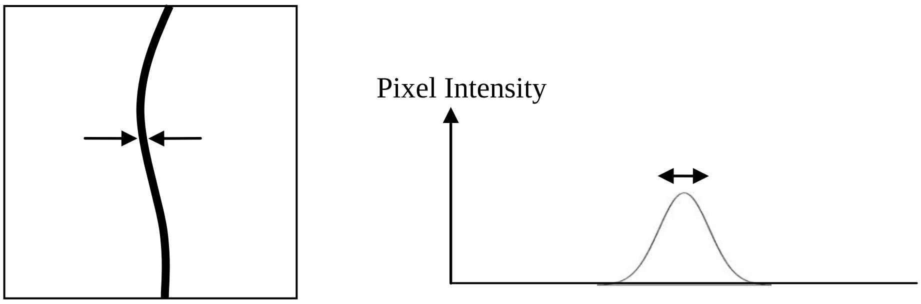

Figure 5. Since no lens is perfectly sharp, pixel intensities are spread by the camera lens. So the tether which may be sub-pixel in diameter may appear 1 or 2 pixels wide in the image.

In many computer-vision systems which involve feature (edge and corner) detection, it is common for the image to be first smoothed using a Gaussian filter to remove the presence of high frequency noise. However in this case, because the tether is so small and will be barely visible in an image, it is high frequencies that are of interest. Smoothing the image with a Gaussian filter would help remove noise from the image, but may also risk losing the tether as well.

Greater resolution can be obtained using a stills camera set up to take a sequence of photographs at regular intervals, rather than video. A commercially available DSLR may have a resolution of 18 megapixels or so (5184px by 3456px) but this is still unlikely to be sufficient to cover the whole 1km length of tether, as the width of the tether would still only subtend roughly 1/20th of a pixel.

The problem of using a camera with a very high resolution is that it greatly increases the amount of data being captured and hence increasing the time spent for image processing. It was found that with large 18 megapixel images, Matlab would often struggle to load and process the images, sometimes resulting in Matlab running out of memory and throwing out an error. For an idea of the amount of data that would be involved, say, for example, 1 hour of footage of the motion of the balloon-tether has been captured using a single 18 megapixel camera taking photographs at 1 second intervals. If the images are captured as jpegs:

18 megapixels per frame – 3 bit/pixel JPEG

Therefore, this equates to 3600 x 18 x (3/8) = 2.43 x 10^7 bytes = 24.3 Gb

It is proposed that multiple cameras can be used as shown schematically in Fig. 3; each camera equipped with a telephoto lens zoomed in to a separate section of the tether. Using as many cameras as required to provide sufficient resolution to be able to detect the tether, which is depended on the perforamance of the tether detection software, the separate images or the extracted tether position data can then later be stitched together to cover the entire length of tether.

Clearly there is a trade-off between the number of cameras used to cover the complete 1km length of tether and the resolution that is achievable. Using more cameras will result in much more data being acquired and further increasing the processing time.

The camera model

Firstly, it is useful to understand how a camera can behaves. The pin-hole camera model is adopted to describe the behaviour of a camera, as shown in Fig. 6. All rays are assumed to pass through a single point, the camera's optical centre O$_C$, intersecting the image plane at a focal distance f from the camera centre.

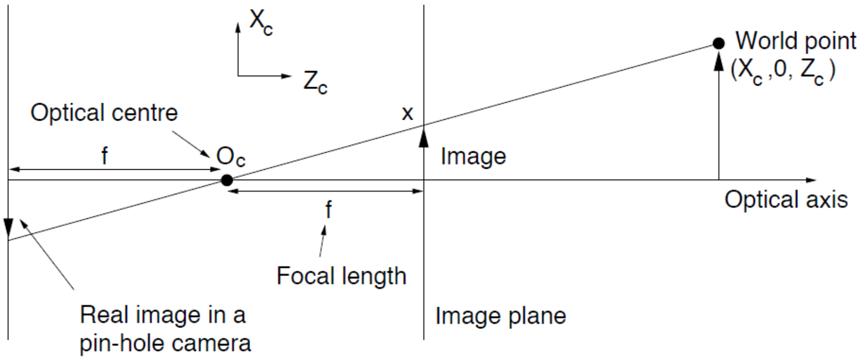

Figure 6. Pinhole camera model.

This is called planar perspective projection. For a ray from a point in the world $\boldsymbol{X_C} = (X_C, Y_C, Z_C)$ intersecting with the image plane at with image coordinates $\boldsymbol{x} = (x, y)$ analysis of similar triangles gives the following relations:

$$x = \frac{fX_C}{Z_C} \qquad y = \frac{fY_C}{Z_C}$$

The CCD array in the camera, as shown in Fig. 7, is modelled by defining pixel coordinates $\boldsymbol{w} = (u, v)$ in addition to the image coordinates $\boldsymbol{x} = (x, y)$, with $\boldsymbol{w}$ and $\boldsymbol{x}$ related as follows:

$$u = u_0 + k_u x \qquad v = v_0 + k_v y$$

where k_u and k_v are scaling factors in the u and v directions respectively and (u_0, v_0) are the coordinates of the optical axis with the CCD array.

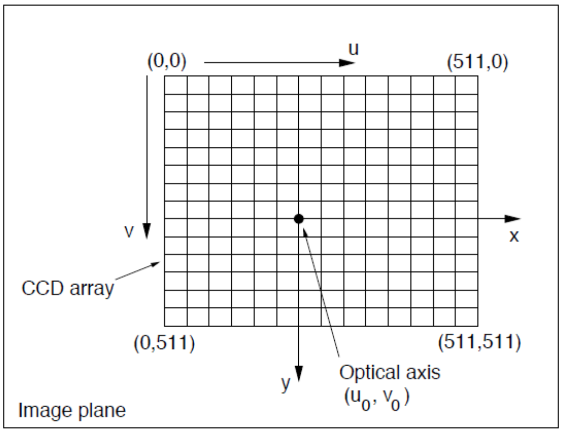

Figure 7. Camera CCD array.

Camera spacing

To observe the entire 1km length of tether with multiple cameras, the angular separation between successive cameras needs to be determined. The distance the cameras are positioned from the base of the tether and the focal length of the lenses required to give sufficient resolution to be able to detect tether will completely determine the number of cameras needed.

For lenses projecting rectilinear (non-spatially distorted) images, as shown in Fig. 8, the effective focal length and the image format dimensions completely define the angle of view. The angle of view can be measured horizontally, vertically or diagonally. With the cameras aligned at 90° (i.e. in a portrait orientation), the angle of view in the horizontal direction is the most important for this application. A portrait orientation is more appropriate to observe a vertical tether, giving maximum coverage in the vertical direction. The angle of view α of a lens producing a rectilinear image is given by Eq. 1[5].

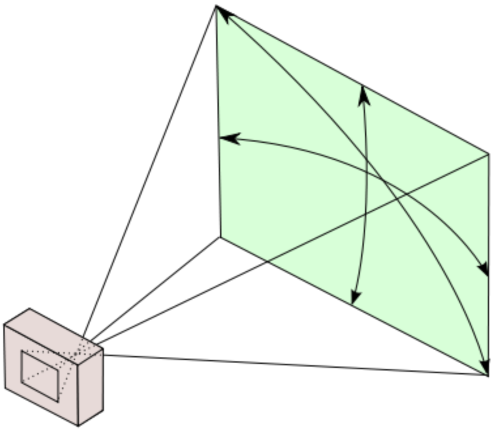

Figure 8. Angle of view of the camera lens projecting a rectilinear image.

$$\alpha = 2\arctan\frac{d}{2f} \tag{1}$$

In Eq.1, d represents the size of the camera sensor in the direction measured and f is the effective focal length. The effective focal length is almost equal to the stated focal length of the lens F, except in macro photography where the lens-to-object distance is comparable to the focal length. In this case, the magnification factor m must be taken into account, Eq. 2.

$$f = F(1+m) \tag{2}$$

For the application of observing the tether, because of the large distances involved, the cameras will be focused at infinity and so the magnification factor can be ignored. For example, consider a Canon 550d camera which has a sensor size of 22.3 x 14.9 mm, with a 100mm telephoto lens. At infinity focus $f = F$ and the angles of view are:

$$\alpha_h = 2\arctan\frac{h}{2f} = 2\arctan\frac{22.3}{2\,x\,100} \approx 12.7°$$

$$\alpha_v = 2\arctan\frac{v}{2f} = 2\arctan\frac{14.9}{2\,x\,100} \approx 8.5°$$

$$\alpha_d = 2\arctan\frac{d}{2f} = 2\arctan\frac{26.8}{2\,x\,100} \approx 15.3°$$

If the cameras are positioned at say 800 metres from the base of the tether, with an angle of approximately 12.7° between each camera, roughly 4 cameras would be needed to cover the complete length of tether, as shown in Fig. 9.

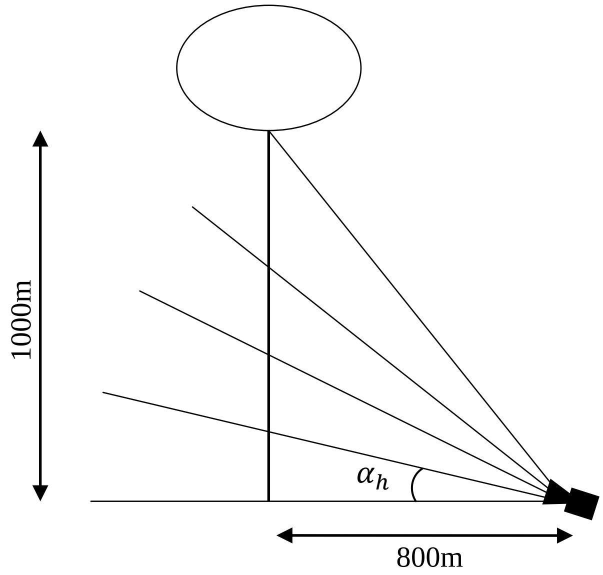

Figure 9. At a distance of 800 metres, 4 cameras with 100mm lenses are required to cover the full length of the tether.

As the cameras become more angled upwards, the cameras observe a greater length of the tether. In order to stitch the images of the separate cameras together, there will need to be a certain amount of overlap between the camera views. It would therefore be more appropriate to use say 5 cameras in this particular configuration to allow for some overlap.

Clearly there is a trade-off between the number of cameras used to cover the full tether and the achievable resolution, i.e. more cameras could be used zoomed-in closer to the tether to give a higher resolution and make the detection of the tether more feasible. The main disadvantage of using more cameras than is necessary, apart from the cost, is that the more zoomed-in the cameras are, the smaller their window of view and therefore the tether is more likely to move out of view of the camera. The angle of view of the camera in the horizontal direction needs to be sufficiently large such that the tether remains in the frame during the motion.

Furthermore, using more cameras will increase the amount of data being captured. This would have repercussions on the time taken to transfer and analyse the data in Matlab, most significantly affecting the time for image processing.

Tether detection methods

The following methods that are described look at extracting the position of the tether from its background, either from video or a sequence of still jpeg images on a frame-by-frame basis. For a captured image sequence, stored in order in a particular file directory, the images are first read into Matlab using the `imread` function placed in a ‘for’ loop over a number frames. On each iteration, a function `findtether` is called to perform the image processing on the image frame, which outputs the extracted x and y pixel coordinates of points along the tether. The code is as follows:

```
for k = startframe:endframe
    filename = [sdirectory '/' jpgfiles(k).name];
    I = imread(filename);

    % Process image "I"
    [x,y] = findtether(I);
end
```

For a video sequence, the code is slightly different. The captured .avi video file is loaded into Matlab using the `mmreader` function, which creates a multimedia reader object, `myVid`. This Matlab object is useful as information such as the total number of frames, the frame rate, image height and width can be extracted, as shown in the following code. A Matlab movie structure is then created from the video frames using a ‘for’ loop and again the `findtether` function is performed to process each frame in turn.

```
myVid = mmreader('filename.avi');
numFrames = myVid.NumberOfFrames
fps = myVid.FrameRate
vidHeight = myVid.Height;
vidWidth = myVid.Width;

% Pre-allocate movie structure.
mov(startframe:endframe) = struct('cdata', zeros(vidHeight,
vidWidth, 3, 'uint8'),'colormap', []);

for k = startframe:endframe
    mov(k).cdata = read(myVid, k);
    I = rgb2gray(mov(k).cdata);
    % Process frame "I"
    [x,y] = findtether(I);
end
```

The `findtether` function contains all that is required to extract the tether position from a single image. Since it is performed on every frame, and a section of recorded footage may contain hundreds of frames, it is important that the code is efficient and not too computationally expensive, otherwise processing the data will become very laborious.

The main principle behind the tether detection algorithm looks at comparing pixel intensities. An image is represented as a matrix $I(x,y)$ of intensity values, for each colour channel (RGB). To examine the pixel intensities more directly, the images can be first converted to gray-scale images, so that each pixel has an intensity value represented by a single number ranging from 0 (black) to 255 (white), either using the inbuilt Matlab function `rgb2gray` or, more efficiently:

```
Ig = 0.3*double(I(:,:,1)) + 0.6*double(I(:,:,2)) +
0.1*double(I(:,:,3));
```

To identify the position of the black tether in an image, one solution is to simply search each entire image frame for the darkest pixels, i.e. pixels with the lowest intensity values, assuming that the tether is darker than its background, which would be mostly sky. This can be done by comparing the pixel intensities, one row of the image at a time, and is easily achieved in Matlab by applying the `min` function to locate the darkest pixel in each row.

```
[z,x] = min(Ig,[],2);
```

The `min` function takes a third argument specifying the dimension in which to search the image (i.e. across the rows or down the columns) and the value and position of the minimum intensity pixels are stored in the variables `z` and `x` respectively. Since the entire image has been searched down the rows, the `y` coordinate values of the tether are simply row numbers. The x and y coordinates of the tether can then be stored for this particular frame and plotted if desired.

Thresholding

In practise the tether may not so easily distinguishable from the background, or the image might be particularly noisy, resulting in incorrect tether detection and so simply searching for the darkest pixels is not sufficient. If any pixels are wrongly identified as part of the tether, these can be rejected by comparing their intensities `z` to a certain threshold value, found by either averaging a section of the background or a known part of the tether. In the code below, the intensities are compared to a calculated threshold intensity value of 120 using the `find` function in Matlab. Only pixels with intensity less than 120 are retained.

```
[z,x] = min(Ig,[],2);
jz = find(z<120);
z=z(jz); x=x(jz);  y=y(jz);
```

This straightforward method works surprisingly well for controlled test footage in the lab, however the method is less successful when the images are very noisy or the tether is particularly faint and difficult to distinguish from the background.

For example, a single frame of video from the motion of a hanging cable set against a plain white wall background is shown in Fig. 10. The original gray-scale jpeg image is shown on the left and the extracted pixels are plotted on the right. Simply searching for the darkest pixels has failed to detect the tether position accurately, identifying pixels which are not part of the cable. This is because it is quite a noisy image and there are regions of the background which are darker than the cable. After applying a threshold to reject pixels below a certain value, a much cleaner result is obtained, as shown in Fig. 11. This result follows the cable much better, although there are still a few pixels wrongly selected.

Although thresholding is a very useful technique and can be very effective, the method will struggle if the tether is set against a more complicated background rather than a plain white background. The method also requires some existing knowledge of the image, i.e. whether a certain part of the image is background or tether, in order to calculate a threshold value. Clearly a more sophisticated approach is required.

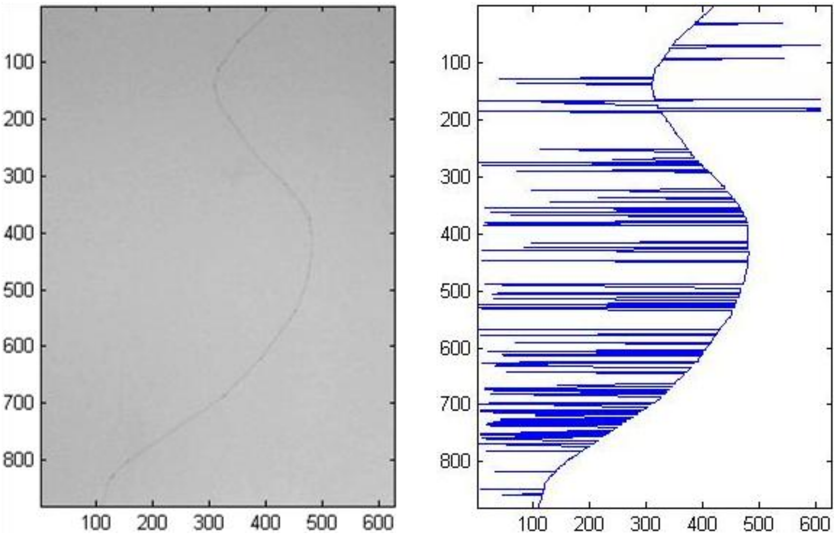

Figure 10. The gray-scale image of a hanging cable is shown on the left. The result after simply searching frame for the darkest pixels is shown on the right.

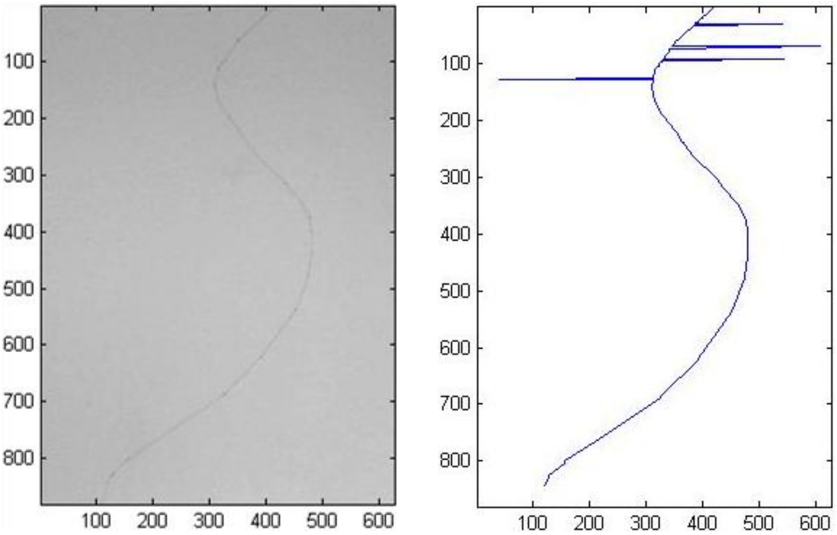

Figure 11. Result after applying thresholding, rejecting pixels below a certain intensity value.

Spatial-temporal processing

Another tether detection method uses the same principle of searching for the darkest pixels, but also makes use of spatial and temporal information by assuming that the tether does not move significantly from one frame to the next. If the tether position is known accurately in one frame, the search for the tether in the following frame can be reduced to a smaller region either side of its previous position. For example, suppose in one frame the tether is in the following position for a region of the image, represented by the matrix I_1:

$$I_1 = \begin{bmatrix} 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix}$$

In this case the tether has the following x-coordinates $x_1 = (3, 3, 4, 5, 6)^T$. And suppose that in the next frame I_2 the tether has moved slightly to another position:

$$I_2 = \begin{bmatrix} 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \end{bmatrix}$$

Rather than searching the entire image for the darkest pixels on every frame, a smaller region either side of the previous tether position can be specified. For this example consider a region of the second frame I_2 that is 1 pixel either side of the tether position in the previous frame I_1, as shown below:

$$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 10 & 0 \\ 0 & 10 & 0 \\ 10 & 0 & 0 \end{bmatrix}$$

Now the `min` search can be applied to this smaller region of the image, which still contains the tether. Within this smaller region the x-coordinates of the tether are $(1, 2, 2, 2, 1)^T$, so to find the true position of the tether in the image, the x-coordinates need to be shifted by the correct amount, so that the tether shows up correctly.

This method has an advantage in terms of efficiency because the entire image does not have to be searched on every frame, and hence greatly reducing the amount of time to process an individual frame, particularly for very large images. This method can also reduce the detection error since only pixels close to where the tether was previously are considered. This means that regions of the image that may actually contain darker pixels than the tether are not searched.

However, problems can arise when using the previous frames to predict the future position. For example, if the tether position in one frame is not known accurately, the algorithm can diverge wildly from the correct position. In the second frame it will use the previous incorrect position to predict a possible region for the future position, which may or may not actually include the tether. So essentially, the prediction for the subsequent position becomes worse after each iteration. The method also needs to take into account the frame rate and the relative speed of the tether, since the expected displacement from one frame to the next needs to be estimated in order to produce a region for the possible position of the tether in the next frame. For example, if the frame rate is only 1 fame per second, the tether could move a significant distance from one frame to the next, so clearly just searching one pixel either side of its previous position will not work. At higher frame rates the tether will move less between sequential frames and therefore the region either side of the tether can be reduced in size.

Filtering

A more sophisticated method is proposed which, rather than using the previous frame to predict the next position, attempts to first improve the signal-to-noise ratio for each individual frame.

Firstly, the image is filtered across the rows by doing the convolution of the gray-scale image with the complex filter `h` as defined below and plotted in Fig. 12. The real part of the filter is plotted in blue and the imaginary part in red. A filter of this shape should help pick out a blip in intensity hopefully corresponding to the tether in each row of pixels, whilst suppressing the surrounding pixel intensities. Hence the main peak of the filter is chosen to be about one or two pixels wide, i.e. of similar width to that expected of the tether. The advantage of using a complex row filter such as this is that it gives phase information about the pixels which can be later used to help determine the angle of the tether.

```
% Define complex row filter.
h = [0 + 0.0014i
   0.0015
        0 + 0.0097i
  -0.0061
        0 + 0.0364i
  -0.1224
        0 - 0.2142i
   0.2541
        0 + 0.2142i
  -0.1224
```

```
          0 - 0.0364i
  -0.0061
          0 - 0.0097i
   0.0015
          0 - 0.0014i];
```

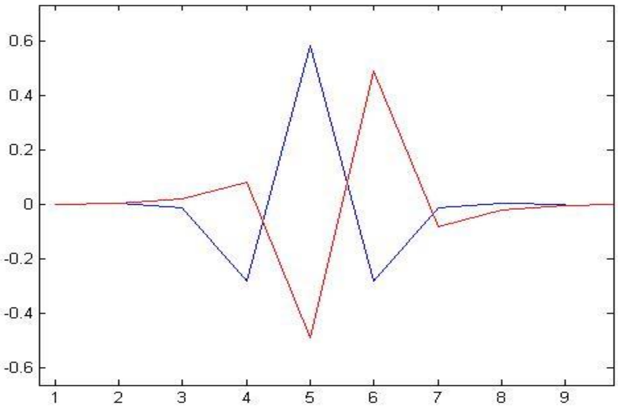

Figure 12. Complex row filter – real part shown in blue and imaginary part in red.

The even rows of the image are convolved with the real part of the filter, and the odd rows are convolved with the imaginary part of the filter. To do this, the complex row filter can be first separated into its real and imaginary parts (`hr` and `hi`) and the image separated into odd and even rows of pixels (`Io` and `Ie`). Then, the inbuilt Matlab function `conv2` is used to perform the convolution separately on the odd and even rows. This is visualized in the diagram in Fig. 13. The convolution with the real part of the filter results in a real output, and the convolution with the imaginary part gives an imaginary output. The real and imaginary outputs can be combined using the `complex` function in Matlab as shown in the code below.

```
Ic = complex(conv2(Ie,hr,'valid'),conv2(Io,hi,'valid')).';
```

The resulting image array `Ic` is half the number of columns of the original image, but now each pixel is represented by a complex number. When converting back to pixel coordinates, one must remember to multiply by 2 so that the tether shows up correctly.

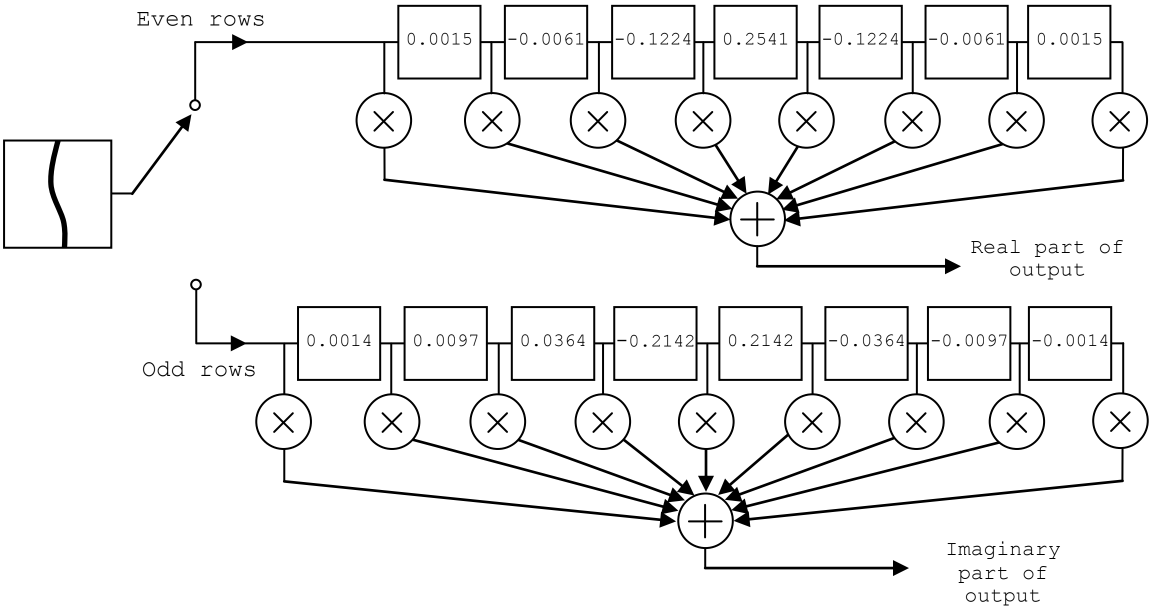

Figure 13. Even and odd rows of pixels are convolved with the real and imaginary parts of the complex filter respectively.

The tether can then be detected at this stage by searching for the maxima in the rows of the image, looking at the magnitude of the pixel values, `abs(Ic)`. As mentioned in the previous section, thresholding can also be applied at this stage to aid in detecting the tether.

However, the signal-to-noise ratio can be improved further by secondly filtering down the columns of the image to increase the pixel intensity in the direction of the tether. If the tether was exactly vertical, pixels could be simply filtered in a vertical direction down the image. However, the tether will be at some unknown orientation which will change along its length. So the image can be filtered in a number of different orientations and the orientation that produces the highest signal-to-noise ratio can be chosen for a particular section of the tether. This is achieved by convolving the image with complex directional filters aligned at angles ranging from -45° to 45°.

The directional filters were designed to be 8x5 arrays and chosen to be aligned in 13 different orientations: in 7.5° increments from -45° to 45°. Any further than 45° then it essentially becomes row filtering again rather than column filtering. It was thought that 7.5° increments would provide sufficient angular resolution. For example, the filters for -45° and 0° are shown in the matrices below. Since matrices are discrete, the complex values mean that the angled filters can still be represented. The 13 directional filters are stored in Matlab as an 8x5x13 array.

$$h_{-45^\circ} = \begin{bmatrix} 0 & -0.2258 - 0.1995i & 0.6298 + 0.5566i & 0 & 0 \\ 0 & -0.1909 - 0.1085i & 0.7741 + 0.4401i & 0 & 0 \\ 0 & -0.1279 - 0.0410i & 0.8921 + 0.2860i & 0 & 0 \\ 0 & 0 & 0.9746 - 0.1011i & -0.0454 + 0.0047i & 0 \\ 0 & 0 & 0.8921 - 0.2860i & -0.1279 + 0.0410i & 0 \\ 0 & 0 & 0.7741 - 0.4401i & -0.1909 + 0.1085i & 0 \\ 0 & 0 & 0.6298 - 0.5566i & -0.2258 + 0.1995i & 0 \end{bmatrix} \qquad h_{0^\circ} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Now every block of 8 rows of the image is convolved with the 8x5 directional filter for each angle. For a generated test image of the arc of a circle, the output after filtering at each angle is shown in Fig. 14. The outputs have been scaled so that they show up correctly when plotted.

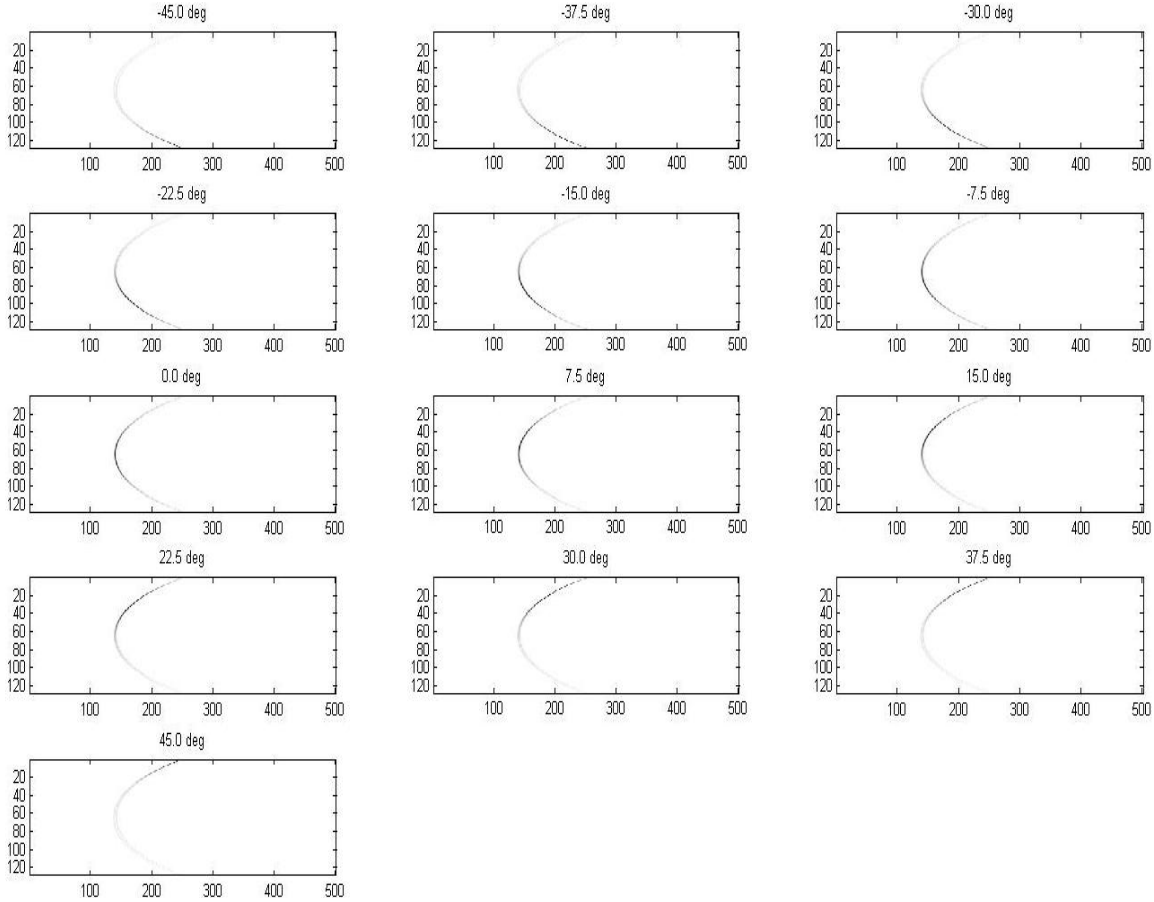

Figure 14. Output of the convolution of a test image of the arc of a circle with 13 directional filters at angles from -45° to 45°.

The arc of a circle image in Fig. 14 shows how convolving the image with filters at different angles has picked out different parts of the arc. When the filter is aligned with the direction tangent to the arc, the output is stronger. The pixels of greater intensity appear darker. If the image of the tether is filtered in this way, the pixels of greatest intensity can then be extracted by choosing the filter appropriate for the orientation of each section of tether.

To find the correct angle of filter for a particular part of the tether, the Radon transform can be utilised. The Radon transform is able to transform a two dimensional image with lines into the domain of possible line parameters, where each line in the image will give a peak positioned at the corresponding line parameters. The Radon transform has lead to many line detection applications within image processing and computer-vision.

The Radon transform for a set of parameters (r, θ) is defined as the line integral through the image $I(x, y)$, where the line is positioned corresponding to the value of the parameters (r, θ). It can be defined mathematically as in Eq. 3[6]. Integrating over the entire image, the magnitudes can be accumulated into a histogram which will show a peak at the parameters defining a line in the image. This is illustrated in Fig. 15.

$$\check{I}(r, \theta) = \int \int I(r, \theta)\, \delta(r - x \cos\theta - y \sin\theta)\, dx\, dy \tag{3}$$

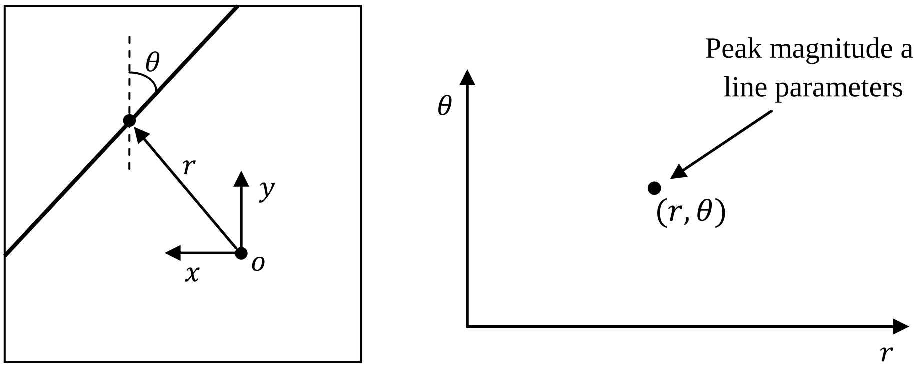

Figure 15. Radon transform can be used to find the presence of a line in an image.

So after filtering in the 13 different directions, for every 8 rows of the image at a time, the pixels can be summed in each direction from -45° to 45° and the resulting magnitudes binned into a histogram, as in the Radon transform. To sum the pixel intensities in each direction, the pixels can be first shifted horizontally by an amount corresponding to the angle of interest θ and then summed vertically, as illustrated in Fig. 16. The summed values are then put into a histogram of x (column position) against angle θ (-45° to 45°), for every 8 rows. For the test image of an arc of a circle, the resulting histogram is shown in Fig. 17. Note that the angle goes from 0 to 13, representing the 13 directions from -45° to 45°.

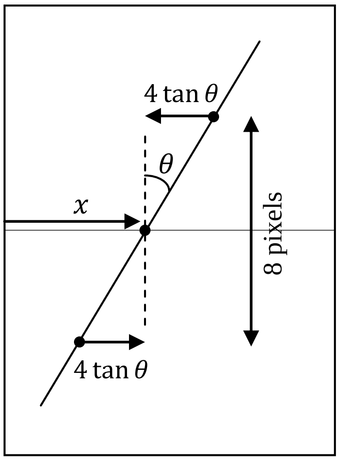

Figure 16. To sum pixels at an angle θ, the pixels are shifted horizontally and then summed vertically. This is repeated for all angles -45° to 45° for every 8 rows of pixels.

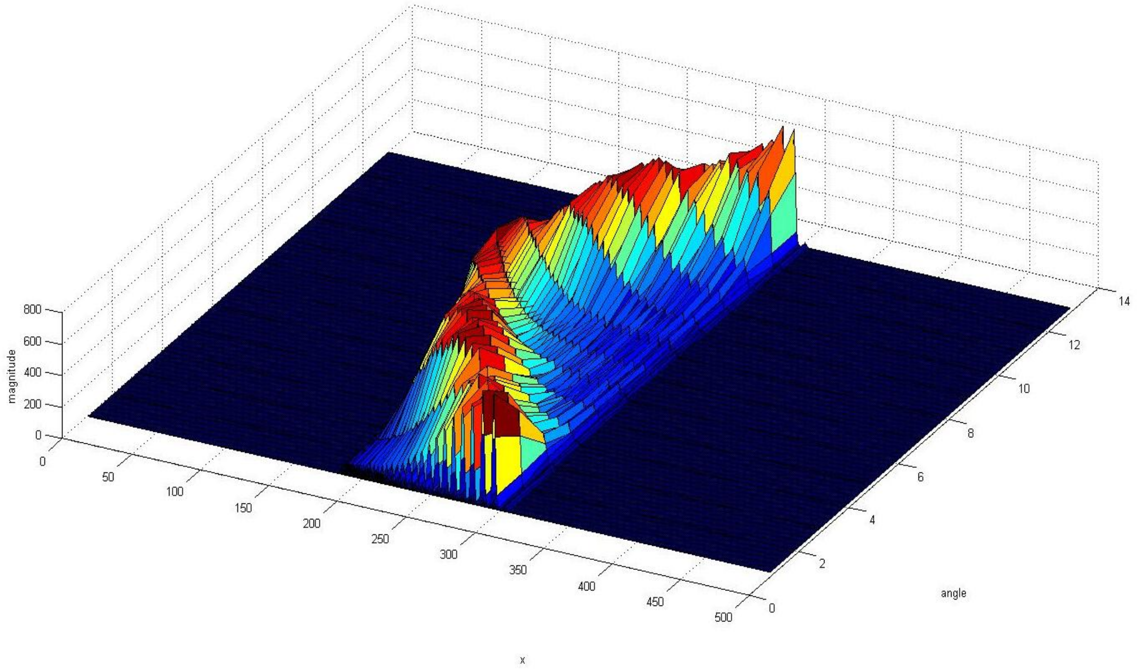

Figure 17. Histogram plot for the test image of an arc of a circle. The pixels in all the columns for every 8 rows are summed in each direction from -45° to 45° and the magnitudes are plotted in the histogram of x (column position) against angle θ (-45° to 45°).

The histogram shows peaks where summing in the direction with parameters (x, θ) agrees with the orientation and position of the arc. The points on the tether can then be extracted by searching for the maxima in the histogram. Because of the convolution with the directional filters, the image must be shifted and scaled so that the extracted tether shows up correctly.

Other constraints can then be implemented such as the assumption that the orientation and position of the tether does not change significantly from one point along the tether to the next (the tether is continuous), or between sequential frames.

For the same test image of a hanging cable as in Fig. 10, this method produces a much cleaner result, as shown below in Fig. 18. The extracted result in the right image is not perfect and some information is lost due to the filtering and convolution, but the extracted pixels follow the shape of the cable much better. However, this method is also more computationally expensive than the other methods discussed, and for a system that contains multiple cameras acquiring many images or frames of video, efficiency is particularly important.

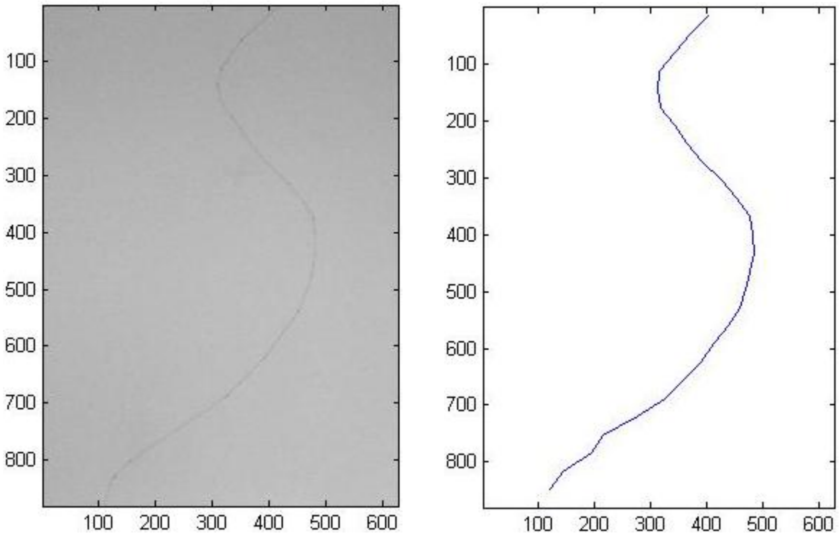

Figure 18. After filtering, the cable can be extracted more successfully.

Camera synchronisation

The use of multiple cameras requires them to be synchronised so that they are triggered simultaneously. Accurate synchronisation in multiple camera systems is important, particularly for a stereo camera system since the position of the tether in space needs to be captured simultaneously in both camera views to obtain an accurate 3D position.

Most DSLR cameras have a remote release socket, as shown in Fig. 19, which allows the cameras to be triggered using a switch. Wiring multiple cameras together and using an intervalometer enables multiple cameras to be fired together at regular intervals. The cameras will not fire at exactly the same time due to the individual shutter lag and the response time of the cameras, but they will usually fire within a few milliseconds of each other[7], which is acceptable if the cameras are triggered at say 1 second intervals. Most DSLR cameras cannot be triggered faster than 1 frame per second continuously due to the buffer capacity and is a limitation of these cameras. Repeated use may also be a concern to the integrity of the mechanical shutter.

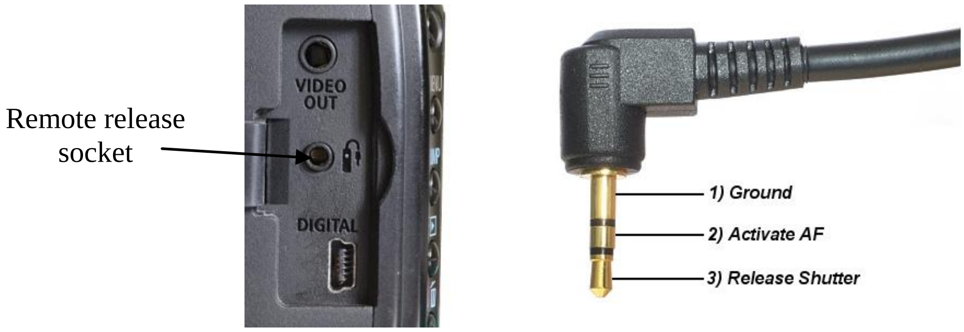

Figure 19. Remote release socket of a Canon DSLR camera and a 2.5mm jack for triggering the camera remotely.

Clearly, the frequency of oscillation of the tether that can be detected depends on the frame rate at which data is sampled, respecting the Nyquist frequency. At 1fps, the maximum frequency of oscillation that can be measured is 0.5Hz. This will just give the overall motion the tether. With video at higher frame rates, higher frequencies can be measured. If the measurement of higher frequencies is necessary, industrial grade (Firewire/GigE) cameras, which provide higher stable frame rates and synchronicity, should be considered.

An alternative method of synchronising multiple cameras could be to use a stroboscope. If the balloon is flown at night, and assuming that the balloon-tether can be illuminated, it could be possible to capture the position of the tether using a strobe effect. With the shutters on all the cameras left open, a strobe light could be triggered to flash at regular intervals. This would avoid the difficulties of trying to synchronise all the cameras electronically. Triggering a single strobe light would be easier, more reliable, and much higher frame rates could also be achieved. The main difficulty using this method is in being able to illuminate the full 1km length of tether. However, this may be an avenue worth pursuing in future work.

Camera calibration

To obtain accurate and meaningful measurements of the tether, the cameras first need to be calibrated against an object of known dimensions – often a grid pattern is used which allows easy identification of individual points at known positions in space. Calibration allows points in the world coordinates system (X_i, Y_i, Z_i) to be related to the pixel coordinates (u_i, v_i) in the image by a single matrix multiplication. This is known as the camera projection matrix P and is a 3x4 matrix in homogenous coordinates[8], as shown in Eq. (4).

$$\underline{w} = P\underline{X} \tag{4}$$

$$\begin{bmatrix} su_i \\ sv_i \\ s \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

The factor s is just a scaling factor. The matrix P has 11 unknowns (since the scale of P doesn’t matter p_{34} can be set to 1). For each point i from the calibration object, the world coordinates (X_i, Y_i, Z_i) and their corresponding pixel coordinates (u_i, v_i) are known, so two equations can be formed each for point as shown in Eq. (5) and Eq. (6).

$$u_i = \frac{su_i}{s} = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \tag{5}$$

$$v_i = \frac{sv_i}{s} = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \tag{6}$$

Since each calibration point gives two equations, to solve for the 11 unknown parameters of the camera projection matrix P, at least 6 known calibration points are needed. Rearranging these equations, a matrix A can be formed as in Eq. (7).

$$A\boldsymbol{p} = 0 \tag{7}$$

$$\begin{bmatrix} -X_i & -Y_i & -Z_i & -1 & 0 & 0 & 0 & 0 & u_iX_i & u_iY_i & u_iZ_i & u_i \\ 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & v_iX_i & v_iX_i & v_iX_i & v_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ \vdots \\ p_{34} \end{bmatrix} = 0$$

The matrix A is a nx12 matrix, where n is the number of known calibration points. The vector $\boldsymbol{p}$ is a 12x1 vector containing the elements p_{ij} of the camera projection matrix P.

This equation can be solved by performing Singular Value Decomposition on the matrix A, using the `SVD` command in Matlab. The values of the projection matrix p_{ij} are the eigenvector corresponding to the smallest eigenvalue of A. However, this linear solution is only approximate because the problem is over-constrained (12 equations and only 11 unknowns) and should be used as a starting point for a non-linear least squares optimisation problem. However, the linear solution works adequately for the level of accuracy that is required for this application. So given the 3D world coordinates for a point in the scene, their corresponding pixel coordinates in the image can be determined.

The projection matrix P can be decomposed into the camera's internal and external parameters, see Eq. 8, by performing QR decomposition on P, using the inbuilt `qr` function in Matlab. The internal camera parameters are stored in the 3x3 matrix K, containing the image scaling factors: where f is the focal length of the lens, (u_0, v_0) is where the optical axis intersects the CCD array, and k_u and k_v are the pixels per unit length in the u-direction and the v-direction respectively. The external parameters are accounted for by a 3x4 matrix consisting of a rotation R and translation $\boldsymbol{T}$, relating the camera coordinate system to the world coordinate system.

$$\begin{bmatrix} su_i \\ sv_i \\ s \end{bmatrix} = K[R|\boldsymbol{T}] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} su_i \\ sv_i \\ s \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

Accurate camera calibration can be very difficult in practise, particularly for outdoor computer-vision systems such as this. The use of a calibration object will be difficult in this situation because most of the cameras will be pointed up into the sky.

To make calibration easier, the number of variables can be reduced by firstly calibrating the cameras in the lab to find the internal camera parameters i.e. the K matrix, which should remain constant for a given focal length. All that is then needed are the external camera parameters, i.e. a matrix governing a rotation and translation transformation. So if the rotation and translation of the camera relative to a world coordinate system can be measured physically, the camera projection matrix P can be calculated.

However, measuring this in practise can prove to be a difficult task. It is difficult to know where the optical centre of the camera is exactly, and it is not easy to measure the precise orientation of the camera in 3D space. It is much more accurate to use a calibration object and recover the projection matrix that way.

One solution is that with many cameras covering the full 1km length of tether, it could be possible to first calibrate the cameras at ground-level using some kind of calibration object or markers, and then to fix the above cameras at known positions and orientations relative to the first cameras. This could be done fairly accurately in the lab, with the cameras fixed in place in a rig. With the ground-level cameras calibrated using a calibration object; it would then be possible to calibrate the above cameras.

A calibration object in the form of a cube of known dimensions was built for calibrating cameras in the lab, as shown in Fig. 20. The points chosen for calibration must not all be coplanar, otherwise all the degrees of freedom of the camera are not being exercised and the resulting equations will not be independent. Fig. 20 shows the calibration cube from two different camera viewpoints. 6 points have been identified in each image; these are marked in blue in the left image and red in the right image. Using the method as described previously, the projection matrices can be found for each viewpoint. The position and orientation of the cameras can be recovered from the projection matrix. The coordinates of the optical centre of the camera can be found using Eq. 9. These have been plotted as shown in Fig. 21.

$$\boldsymbol{c} = -RK^{-1}(p_{14}, p_{24}, p_{34})^T \tag{9}$$

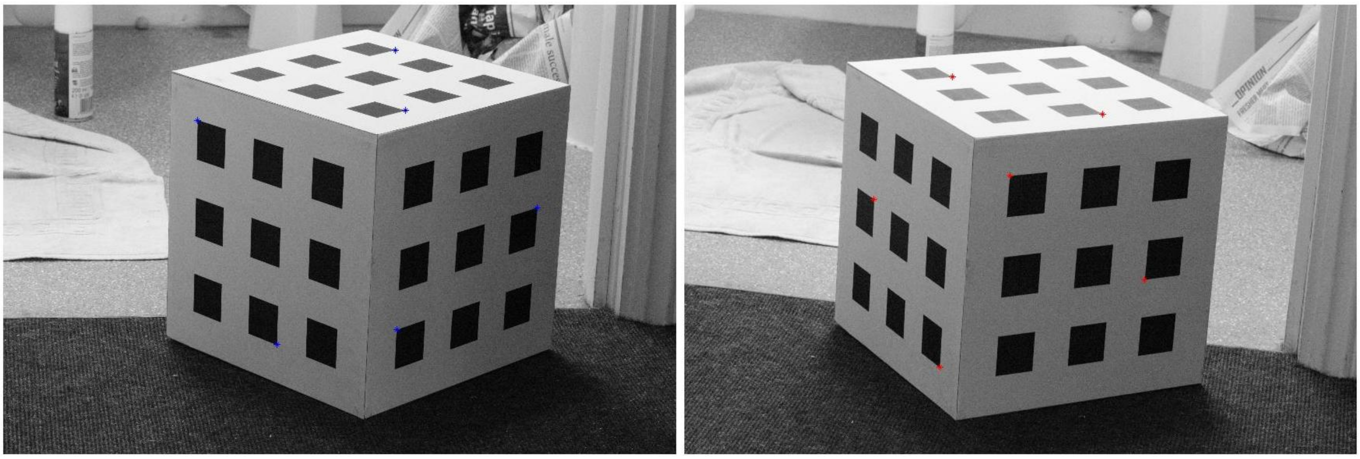

Figure 20. A calibration cube observed from two different camera viewpoints. 6 points for calibration are identified in each image, marked in blue in the left view and red in the right view.

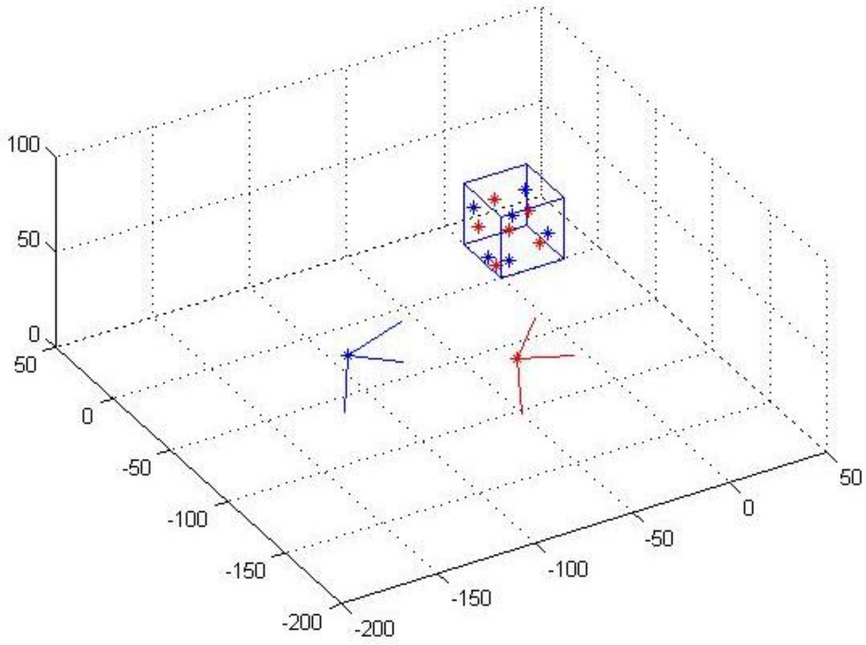

Figure 21. The camera position and orientation can be recovered from the projection matrix for calibrated cameras. The scale is in cm.

Stereo vision

To obtain the 3D coordinates of points along the length of the tether, a stereo camera setup needs to be employed. With a single camera image it is only possible to determine the camera ray on which a feature lies. Using two sets of cameras observing the same feature from two different viewpoints, the intersection of the rays can be determined, see Fig. 22.

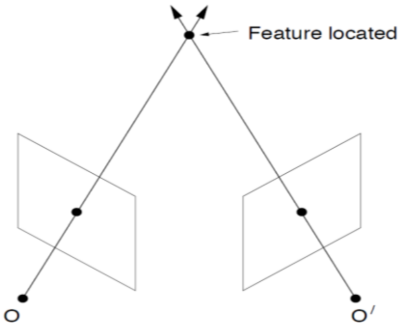

Figure 22. Features can be located in 3D using a pair of cameras in a stereo configuration.

If the cameras can be calibrated with respect to a world coordinate system, i.e. the projection matrices for each camera are known, it is not difficult to recover the 3D coordinates of a feature in space common to both camera views.

To be able to find the intersection of the rays, corresponding points need to be found in both images. The epipolar constraint facilitates this by constraining the search for the corresponding point in the second image to a line as shown in Fig. 23. The epipolar constraint arises from the fact that the two rays from the optical centres of the two cameras to a common scene point lie in a plane called the epipolar plane[3]. The intersection of the epipolar plane with each image plane defines a line called an epipolar line. It is possible to find the equation of the epipolar line in one image, corresponding to a point in the other image. The epipolar constraint reduces the search for the corresponding point from a 2D search over the entire image, to a 1D search along the epipolar line.

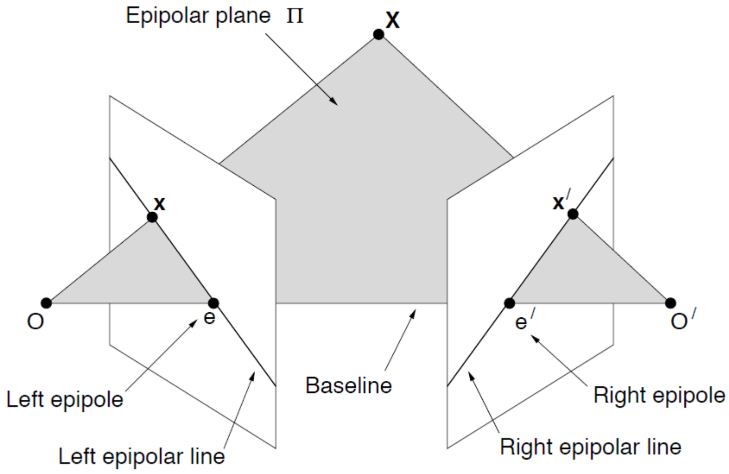

Figure 23. The epipolar plane defines the epipolar constraint.

If the left and right cameras of a stereo system can be related by a rotation R and translation $\boldsymbol{T}$, as shown in Fig. 24, then a point in the world with 3D coordinates $\boldsymbol{X}$ in the left camera's coordinate system and $\boldsymbol{X'}$ in the right camera's coordinate system can be related as in Eq. 10.

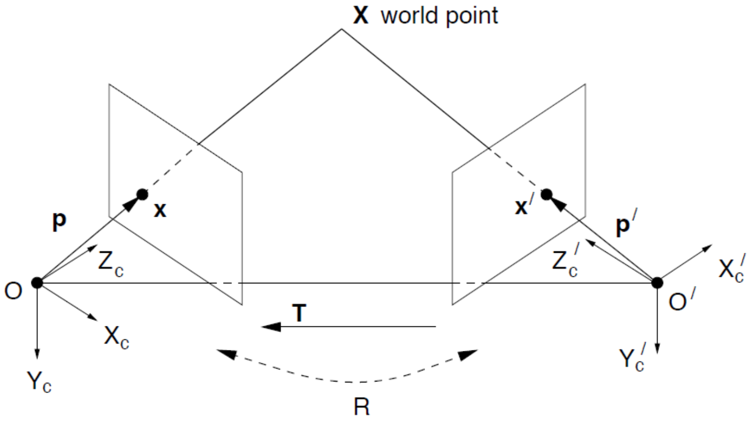

Figure 24. The geometry of a stereo camera configuration, the left and right cameras can be related by a translation and rotation.

$$\boldsymbol{X'} = R\boldsymbol{X} + \boldsymbol{T} \tag{10}$$

Taking the vector product of Eq. 10 with the translation vector $\boldsymbol{T}$ followed by the scalar product with $\boldsymbol{X'}$:

$$\boldsymbol{T} \times \boldsymbol{X'} = \boldsymbol{T} \times (R\boldsymbol{X} + \boldsymbol{T})$$
$$\boldsymbol{T} \times \boldsymbol{X'} = \boldsymbol{T} \times R\boldsymbol{X}$$
$$\boldsymbol{X'} \cdot (\boldsymbol{T} \times \boldsymbol{X'}) = \boldsymbol{X'} \cdot (\boldsymbol{T} \times R\boldsymbol{X})$$
$$\boldsymbol{X'} \cdot (\boldsymbol{T} \times R\boldsymbol{X}) = 0$$
$$\boldsymbol{X'}^{\boldsymbol{T}} E\boldsymbol{X} = 0$$

$\boldsymbol{T} \times R = E$ is known as the essential matrix. This equation holds for the rays $\boldsymbol{p}$ and $\boldsymbol{p'}$ passing through the left and right camera centres, as shown in Fig. 24, so the equation becomes Eq. 11.

$$\boldsymbol{p'}^{T} E\boldsymbol{p} = 0 \tag{11}$$

For the left and write cameras with internal parameter matrices K and K' respectively, the following can be written:

$$\boldsymbol{w} = K\boldsymbol{p} \qquad \Rightarrow \qquad \boldsymbol{p} = K^{-1}\boldsymbol{w}$$
$$\boldsymbol{w'} = K'\boldsymbol{p} \qquad \Rightarrow \qquad \boldsymbol{p'} = K'^{-1}\boldsymbol{w'}$$

Substituting into Eq. 11:

$$\left(K'^{-1}\boldsymbol{w'}\right)^{T} EK^{-1}\boldsymbol{w} = 0$$
$$\boldsymbol{w'}^{T} K'^{-T} EK^{-1}\boldsymbol{w} = 0$$

$$\boldsymbol{w'}^{T} F\boldsymbol{w} = 0 \tag{12}$$

$$[u_i' \quad v_i' \quad 1] \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = 0$$

where $K'^{-T}EK^{-1} = F$ is known as the fundamental matrix, and is useful for defining epipolar geometry. The result in Eq. 12 shows that the pixel coordinates of corresponding points in both views can be related by the 3x3 fundamental matrix. A point in the right view

$\boldsymbol{w}'$ that corresponds to a point $\boldsymbol{w}$ in the left view must lie on the epipolar line $\boldsymbol{l}'$, so the following can be written:

$$\boldsymbol{w}' \cdot \boldsymbol{l}' = 0$$

But from Eq. 12,

$$\boldsymbol{w}' \cdot F\boldsymbol{w} = 0$$

Therefore, the equation of the epipolar line is given by,

$$\boldsymbol{l}' = F\boldsymbol{w}$$

And the epipolar line in the left image is,

$$\boldsymbol{l} = F^T\boldsymbol{w}'$$

The left and right epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$ are defined as the point in each image which is common to all the epipolar lines and are given by the null spaces of F and F^T respectively.

$$F\boldsymbol{e} = 0$$

$$F^T\boldsymbol{e}' = 0$$

The following explains how the fundamental matrix can be calculated in practise. If the projection matrices P and P' for the left and right cameras are known, it is possible to recover the fundamental matrix and hence obtain the epipolar geometry. The position of the optical centre of the left camera $\boldsymbol{c}$ can be computed from the projection matrix,

$$\boldsymbol{c} = -RK^{-1}(p_{14}, p_{24}, p_{34})^T$$

This can be represented in homogenous coordinates by a 4-vector $\boldsymbol{C} = (\boldsymbol{c}^T 1)$ so that,

$$P\boldsymbol{C} = \boldsymbol{0}$$

and its projection into the right image plane defines the epipole $\boldsymbol{e}'$**,**

$$\boldsymbol{e}' = P'\boldsymbol{C}$$

Because the projection matrix for each camera is a 3x4 matrix, it is necessary to compute the pseudo-inverse P^+ of the projection matrix P,

$$P^+ = P^T(PP^T)^{-1}$$

such that multiplication with the projection matrix of the left camera gives the identity matrix I, and multiplication with the projection matrix of the right camera gives a 3x3 matrix (a two-dimensional projective transformation) M:

$$I = PP^+$$
$$M = P'P^+$$

The fundamental matrix F is then given Eq. 12.

$$F = \boldsymbol{e}' \times M = \boldsymbol{e}'_\times M \qquad (13)$$

This can be implemented in Matlab as follows:

```
P1plus = P1'*inv(P1*P1');
M = P2*P1plus;
F = [0,-e2(3),e2(2);e2(3),0,-e2(1);-e2(2),e2(1),0]*M;
```

If the cameras are not calibrated and the projection matrices are not known (i.e. E, K and K' are not known a priori), the fundamental matrix can still be estimated from point correspondences between images.

The two images for the left and right cameras of a stereo system observing a calibration object are shown in Fig. 25. The point identified in red in the left image is mapped to the blue line in the right image by the epipolar constraint. Every point along the blue line will intersect with the ray from the left image. All that remains is to search along the blue line to find the correct matching point in the right image.

This technique can be applied quite easily to the tether. Once the tether position has been extracted from its background using the methods given in the previous sections, for each data point in one image the equation of the constraint line is found in the other image. Then, the point on the tether that intersects with the constraint line needs to be found. Since the tether is one-dimensional, there is only one possible point of intersection.

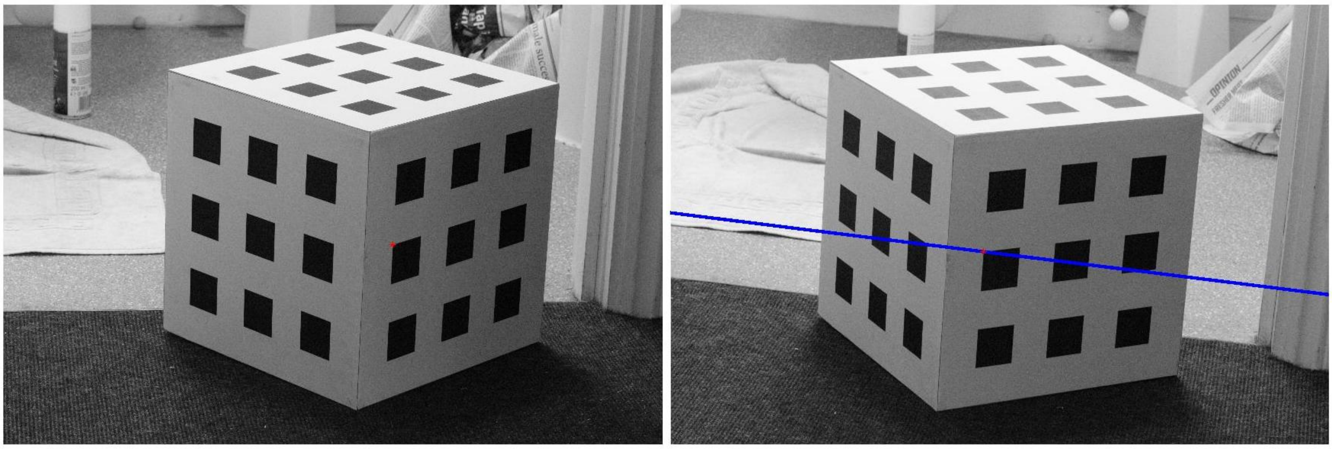

Figure 25. Left and right views of a stereo camera setup, viewing the calibration object. The point marked in red in the left image is mapped to the blue line in the right image by the epipolar constraint.

This can be applied to the example of a hanging cable. The left and right camera views for a particular frame in the image sequence observing the motion of a swinging cable are shown in Fig. 26.

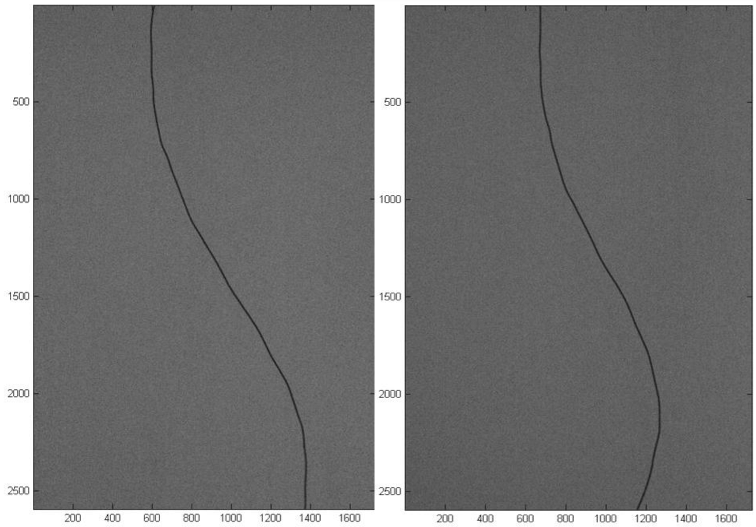

Figure 26. Left and right views of a stereo camera setup viewing the motion of a hanging cable.

The position of the hanging cable is then extracted in the left and right camera views using the detection methods discussed previously, as shown in Fig. 27. Extracted data points on the cable in the left view are shown in red. Calculating the fundamental matrix and using the epipolar constraint, these can be mapped to the red epipolar lines in the right view. The intersections of the lines with the cable are marked in green.

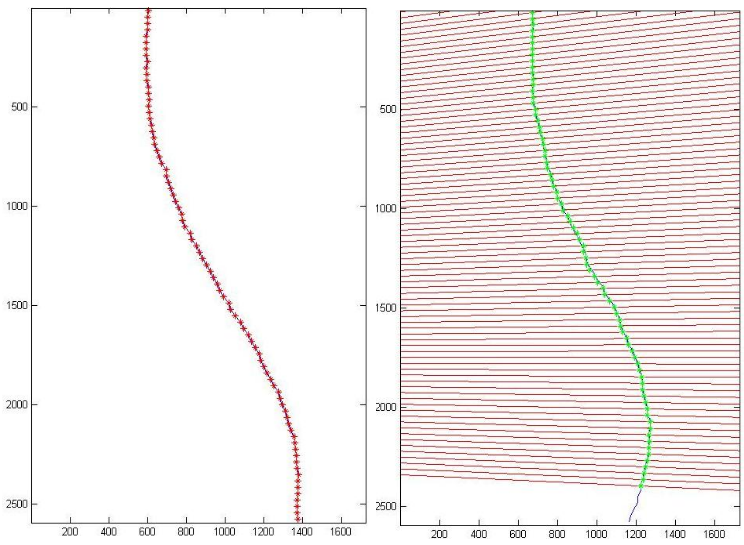

Figure 27. The cable position is extracted in each view. Then each data point marked in red in the left image is mapped to the red lines in the right image by the epipolar constraint. The intersections of these lines with the cable are shown in green.

This process is repeated for all data points, so the image coordinates of all points along the cable in the left image (u, v) and their corresponding coordinates in the right image (u', v') are known. With the coordinates of each corresponding point in both images, the 3D world coordinates are obtained by triangulation; by essentially solving the simultaneous equations given by the projection matrix relation in Eq. (4). For each pair of corresponding points, the following can be written:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = [P_{ij}] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \qquad \begin{bmatrix} u_i{}' \\ v_i{}' \\ 1 \end{bmatrix} = [P_{ij}{}'] \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

This gives 4 equations and 3 unknowns and can be solved by rearranging into the form $A\boldsymbol{X} = 0$ and performing Singular Value Decomposition on the matrix A, using the SVD function in Matlab. The complete 3D construction of the hanging cable is plotted in Fig. 28. The result is accurate and to scale (in centimetres) since the cameras were calibrated beforehand. The advantage of this method is that no identifiable markings on the tether are required.

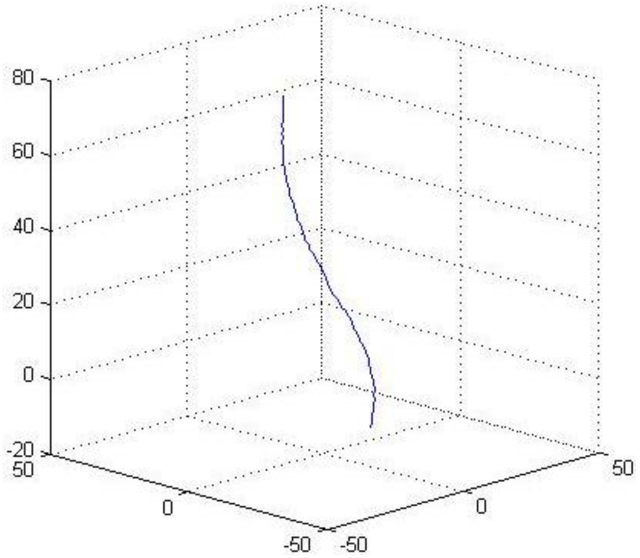

Figure 28. The complete 3D reconstruction of the hanging cable (scale in centimetres).

Point tracking

One significant advantage of the tether detection system designed is that no external markings on the tether are required to find its position in space. The cameras are simply pointed at the unmarked tether, data is gathered, and the 3D coordinates of points along the tether are extracted in each frame.

However, if the motion of a single point on the tether is to be tracked over time, there needs to be some way of identifying the position of a specific point on the tether, over a number of consecutive frames.

One method of doing this would be to integrate to a certain length along the tether from a fixed or known point on the tether (e.g. the fixed base point of the tether) using the equation for calculating the length of arc in 3D, as given in Eq. 14. This requires that a parametric equation describing the position of the tether in 3D is known.

Instead, it may be easier to use the equation for the length of an arc in 2D, as given in Eq. 15, to find the position of a point in the left and right camera views, and then to find its 3D coordinates. An equation for the tether in the 2D views can be found approximately by fitting a polynomial to the tether using the `polyfit` command in Matlab. This approximate equation describing the tether can then be integrated.

So if the world coordinates of the base of the tether are known, then it should be possible to find the position of a point say 100m along the tether in every frame and therefore track its motion over time.

$$L = \int_C ds = \int_{t_1}^{t_2} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}\, dt \tag{14}$$

$$L = \int_C ds = \int_a^b \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx \tag{15}$$

However, this method of integrating a certain length along the tether assumes that the tether is inextensible. The tether is actually designed to be 800 metres in length and stretches to 1km. Therefore, this assumption may not be valid.

If this method is not possible, identifiable markings may have to be used on the tether after all. It could be possible to put white markings every 50 metres or so along the 1km length of tether. The tether detection algorithm will then, as before, search for the darkest pixels, but will find drops in pixel intensity along its length corresponding to the white markings. With simple thresholding, these pixels can be identified and their positions tracked over time.

The System

The previous sections have discussed the theory and design methods of creating a system capable of detecting and tracking the motion of the tether. The different elements of the system, from image processing to 3D reconstruction using stereo cameras, need to be combined. The main structure and requirements of the complete system are listed below. In Matlab, separate functions have been created to perform the separate tasks.

1. Capture data using multiple cameras.
2. Calibrate cameras.
    - Load calibration image.
    - Locate image coordinates u, v of at least 6 calibration points.
    - Determine corresponding world coordinates X, Y, Z with reference to some origin.
    - Solve for projection matrix P for each camera.
    - Calculate fundamental matrix F.
    - Output P_1, P_2 and F.

3. Load captured image sequence/video data of tether motion.
4. Tether detection.
   - For each image/video frame call tether detection function.
   - Output extracted u, v data for sampled points along the tether.
   - Repeat for second camera u', v'.
5. 3D reconstruction.
   - Use epipolar geometry for find corresponding image coordinates.
   - Solve for world coordinates of each point on the tether.
   - Output X, Y, Z for each point along tether.
6. Image stitching.
   - Combine sections of the tether from multiple cameras.
7. Point tracking.
   - Track the motion of specific points over time.

4. Experimental testing

Small-scale tests were carried out in the lab using hanging chains and helium balloons attached to strings. Larger-scale outdoor tests using kites were performed in an effort to recreate the conditions of the actual full-scale balloon launch as close as possible.

Motion of a hanging chain

To test the tether detection algorithm, the 2D motion of a hanging chain was observed using a single camera. The chain was attached to a white wall at one end and the other end was displaced by a small amount and then released. The subsequent motion was captured on video at 240 frames per second.

The chain is extracted in each frame by searching for the darkest pixels. Because of the high frame rate, the chain does not move significantly from one frame to next. So using a combination of the spatial-temporal processing and thresholding methods described earlier, the chain is able to be picked out against the background. This is shown for a selection of frames in Fig. 29

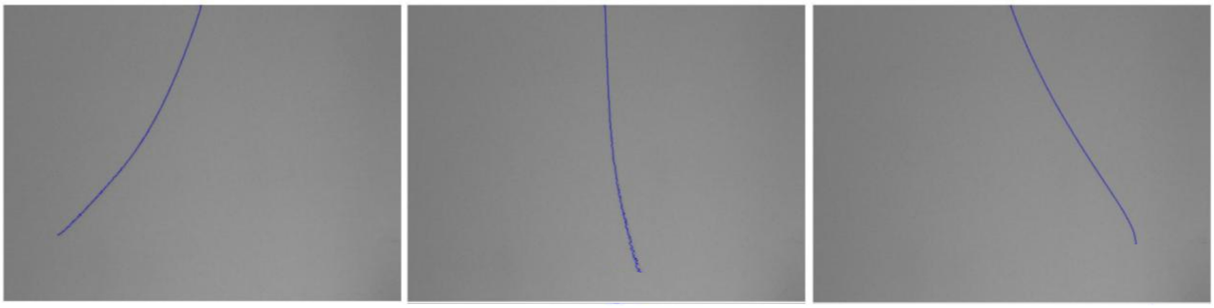

Figure 29. Frames from the motion of a hanging chain. The position of the chain has been successfully extracted in each frame.

A curve can be fit to the shape of the chain in each frame using the `polyfit` function in Matlab. Then, using the formula for calculating the length of an arc given in Eq. 15, the curve can be integrated to a certain length in each frame, to track the position of a specific point on the chain over a number of frames. With a frame rate of 240 fps, the time interval between successive frames is 0.0042 seconds, and so a displacement-time graph can be plotted. The velocity and acceleration of points along the chain can also be calculated if desired, as can the frequency spectrum.

Since the chain was displaced by a relatively small amount, the approximate displacement of points along chain can be found by measuring the displacement in a purely horizontal direction. The displacement-time graphs for 3 points along the chain have been plotted, as shown in Fig. 30.

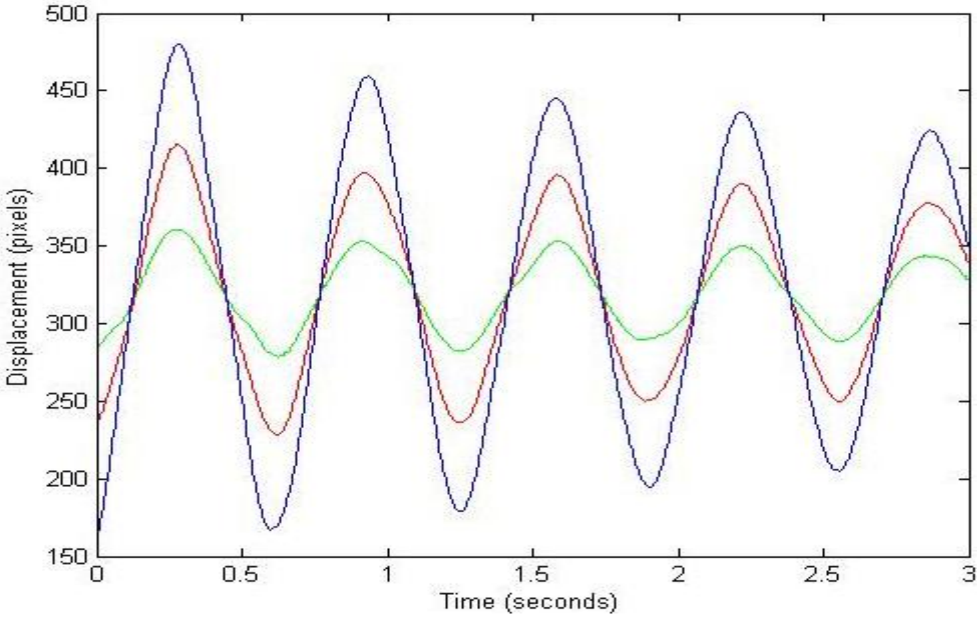

Figure 30. Displacement-time graph for 3 points along the length of the chain.

The period of oscillation is approximately 0.65 seconds, and the frequency is 1.55Hz = 9.74rad/s. The motion of a hanging chain is described by Bessel functions and the frequency of oscillation for the Mth mode is given by Eq. 16, where r_M is the Mth root of the zeroth order Bessel function $J_0(u)$.[9]

$$\omega_M = \frac{r_M}{2}\sqrt{\frac{g}{L}} \tag{16}$$

Given that the length of the chain was roughly L = 0.7m, for the second mode of vibration, $r_M = 5.52$, the frequency of oscillation is:

$$\omega_M = \frac{5.52}{2} \times \sqrt{\frac{9.81}{0.7}} = 10.32 \text{ rad/s}$$

This agrees fairly well with the frequency observed.

Kite testing

It has been shown that it is possible to recover the 3D position of a hanging cable in the lab using stereo cameras for a sequence of images, as in Fig. 28. If it can also be shown that the tether is at least detectable against a sky background, then in theory, it should be possible to find the 3D coordinates of the tether. To test this, experiments using kites were performed.

Camera setup

For experiments using kites, the following apparatus was used, as shown in Fig. 31 and Fig. 32:

- 2 cannon 550d cameras with 70-200mm telephoto lenses
- 2 tripods
- Intervalometer
- Remote release connecting cables
- Kite
- 3mm black kite line

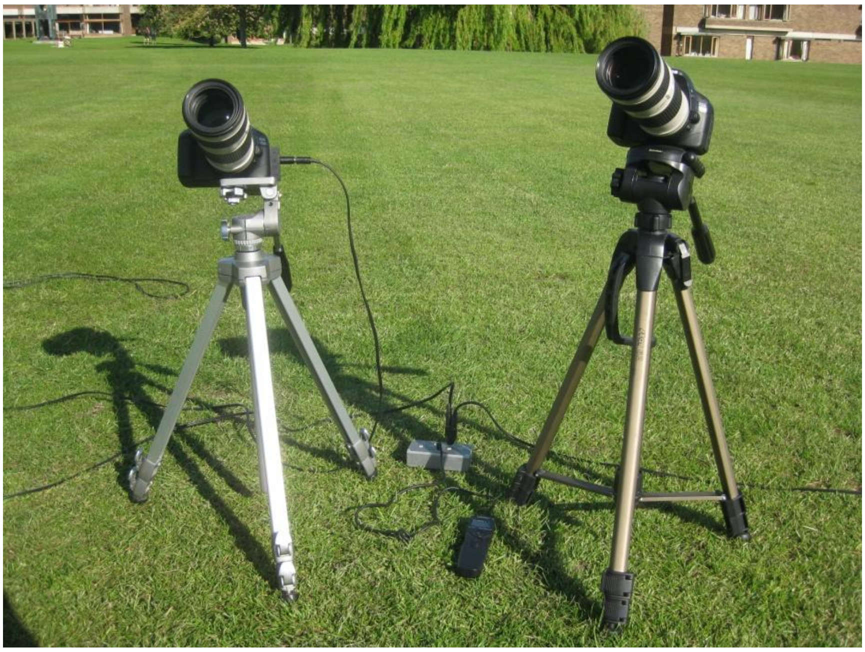

Figure 31. Stereo camera setup for outdoor kite testing.

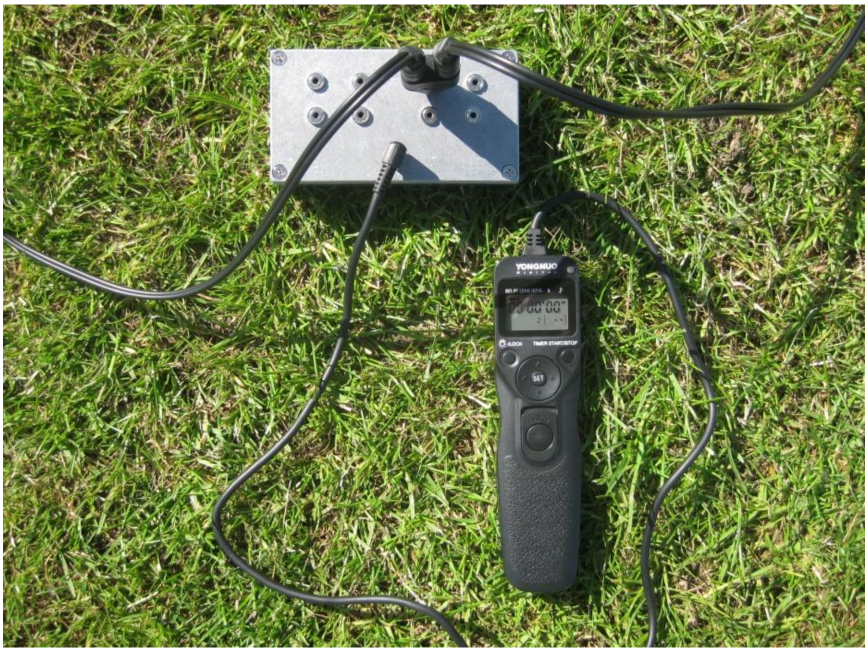

Figure 32. Intervalometer and junction box for synchronised triggering of cameras.

The outdoor tests using kites provide a sky background and variable wind and weather conditions. A 3mm black line on the kite was used to provide sample test footage for testing the tether detection method. The kite was flown to an altitude of roughly 70m. Looking at the sample shot of the kite line in Fig. 33, the line is quite difficult to pick out from the cloudy background, even with the naked eye.

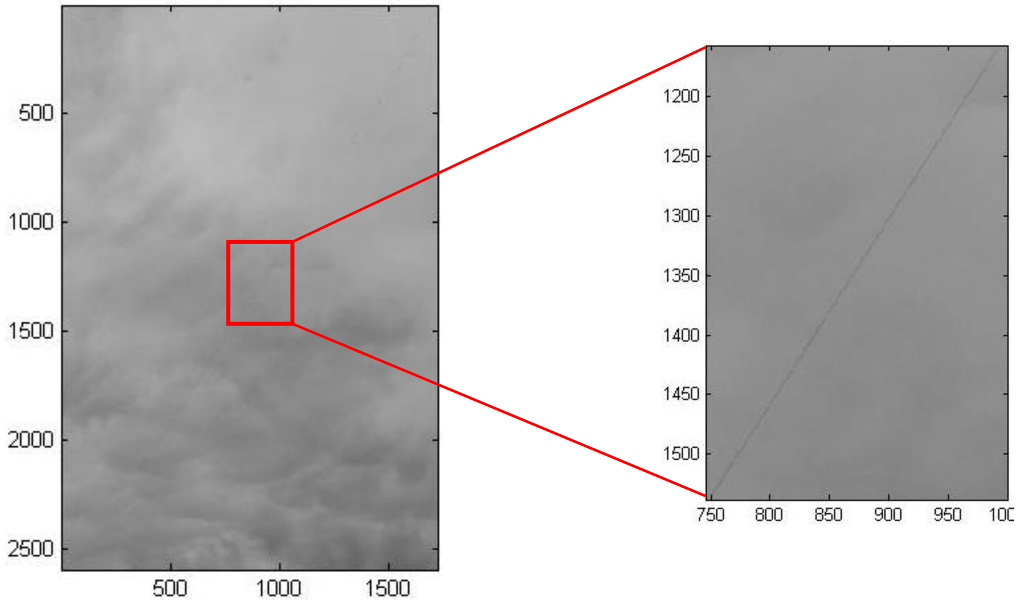

Figure 33. Image of the kite string against a cloudy background.

Looking at a small cross section of pixel intensities across the centre of the image, as shown in Fig. 34, there is a significant drop in the pixel intensity value indicating the position of the kite string at around column 900. This can be identified correctly by performing a simple `min` search across the image.

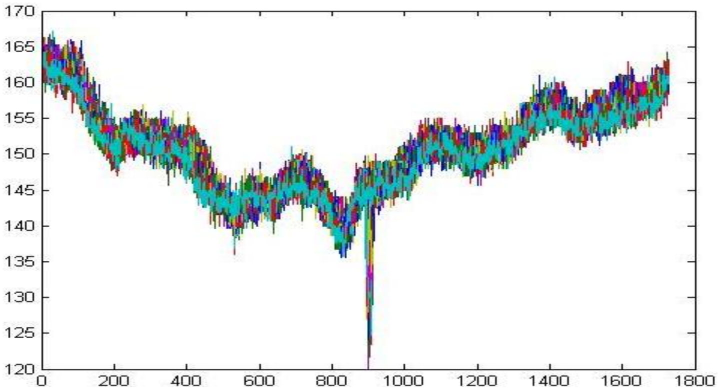

Figure 34. Cross section of pixel intensities across the centre of the image. The drop intensity corresponding to position of the kite string can clearly be identified at around pixel 900.

However looking at the pixel intensities in row 1500 for example, shown in Fig. 35, the pixel intensity value is lower at pixels other than where the tether is, therefore simply searching the image in its current state for the darkest pixels will not work.

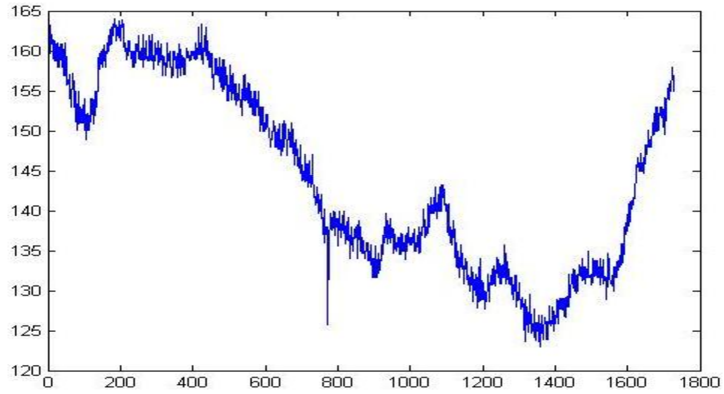

Figure 35. Plot of pixel intensities in row 1500 of the image. The position of the kite string at around pixel 780 is less noticeable since there are pixels with lower intensity elsewhere.

The output after performing convolution with the complex filter is shown in Fig. 36. The filtering method has successfully emphasised the peak corresponding to the kite string and has suppressed the surrounding pixel intensities. Note that the pixel scale is different in both the x and y directions due to filtering, so the result must be shifted by the correct amount in order for the tether to show up correctly when plotted. The peak in intensity can then be detected by searching for the pixels with highest intensity values, using the `max` function in Matlab.

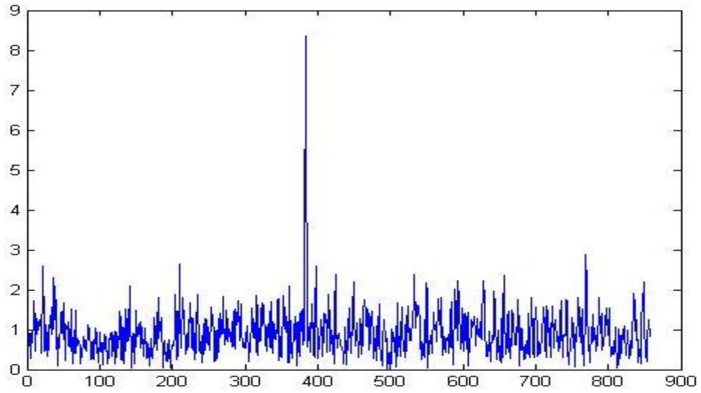

Figure 36. Plot of pixel intensities in row 1500 of the image after filtering. Filtering has managed to emphasise the position of the kite. Note that the scale is different due to filtering.

The kite string can be successfully extracted from the sky background using the filtering method, as shown in Fig. 37. This shows that it is possible to extract the position of the tether over a sequence of frames.

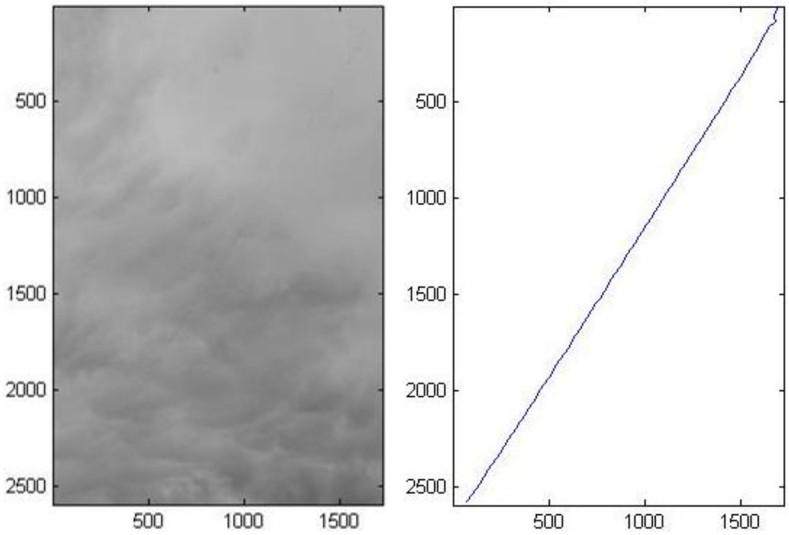

Figure 37. The kite line has successfully been extracted using the filtering method, the original gray-scale image shown on the right and the extracted pixels shown on the left.

The two cameras were then setup in a stereo configuration. The cameras were positioned roughly 10 metres apart and about 30 metres from the base of the kite line. Setting this up provided several challenges and care had to be taken to make sure the two cameras were observing the same section of the kite string. 4 white markings were attached to the string at 20 metre intervals to aid this.

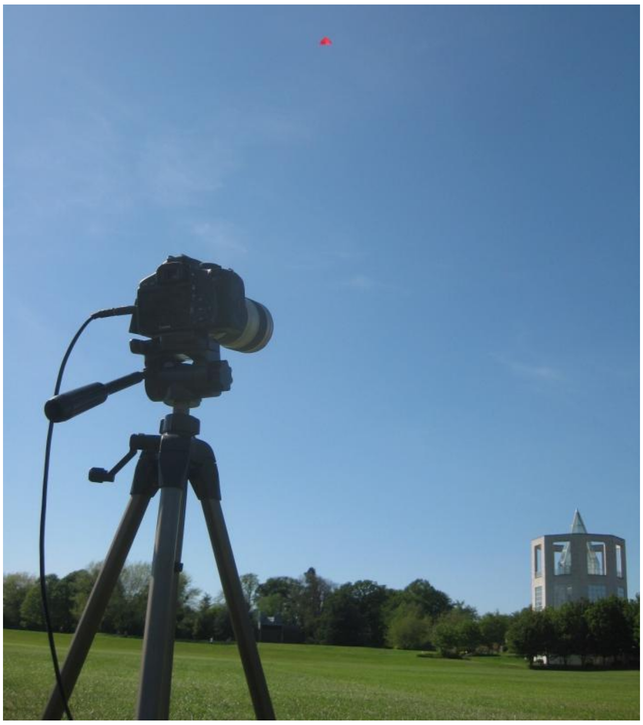

Figure 38. The Moller Centre building shown on the left of the image was used to calibrate the cameras. The red kite can be seen near the top of the image.

The main difficulty was in calibrating the cameras which are pointed into the sky, clearly the calibration cube, as used previously in the lab, would not work in this situation. The kite tests were carried out on Churchill college sports grounds, and it was suggested that the Moller Centre building, shown in Fig. 38, could be used for a rough calibration.

The dimensions of the Moller Centre were obtained through a combination of measurements made by-hand and from acquired architectural drawings. This meant that the world coordinates for at least 6 points on the building could be found, with reference to some arbitrary origin, to enable calibration of the cameras. The cameras were orientated such that they would cover the same section of the kite string, with part of the Moller Centre framed at the bottom of the image. This is shown in Fig. 39 for the left and right camera views. 6 points with known world coordinates are shown and are marked in blue in the left view and red in the right view.

Ideally, an object used for calibration should span most of the image frame to exercise all the degrees of freedom of the camera. Since the building only occupies a small region in the corner of each image, and only 6 calibration points have been identified, the calibration will not be entirely accurate, but will be sufficient to give approximate dimensions. From the calibrated cameras, the position and orientation of the cameras have been recovered and plotted as shown in Fig. 40. The plot shows the cameras are about 10 metres apart and 20 metres below the top of the tower. This roughly agrees with the measurements taken in the field.

Figure 39. The left and right views for the stereo setup observing the kite string. The Moller Centre is positioned at the bottom of the frame to provide calibration. The coordinates of 6 known points in each image have been identified, marked in blue in the left image and red in the right image.

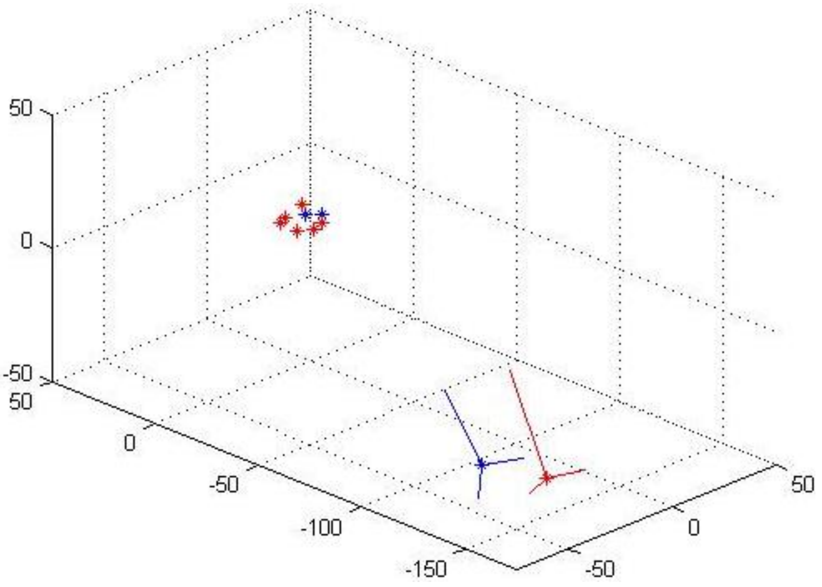

Figure 40. The position and orientation of the cameras have been recovered from the projection matrices for each camera. The scale is in metres.

Cropping the images to just above the building, the kite string is then extracted in each image as shown in Fig. 41. For each data point in the left image, the corresponding epipolar lines are plotted in red in the right image, their intersections with the string are shown in green. Notice that the epipolar lines intersect with only part of the kite string in the right image because the cameras were not observing the exact same portion of the string.

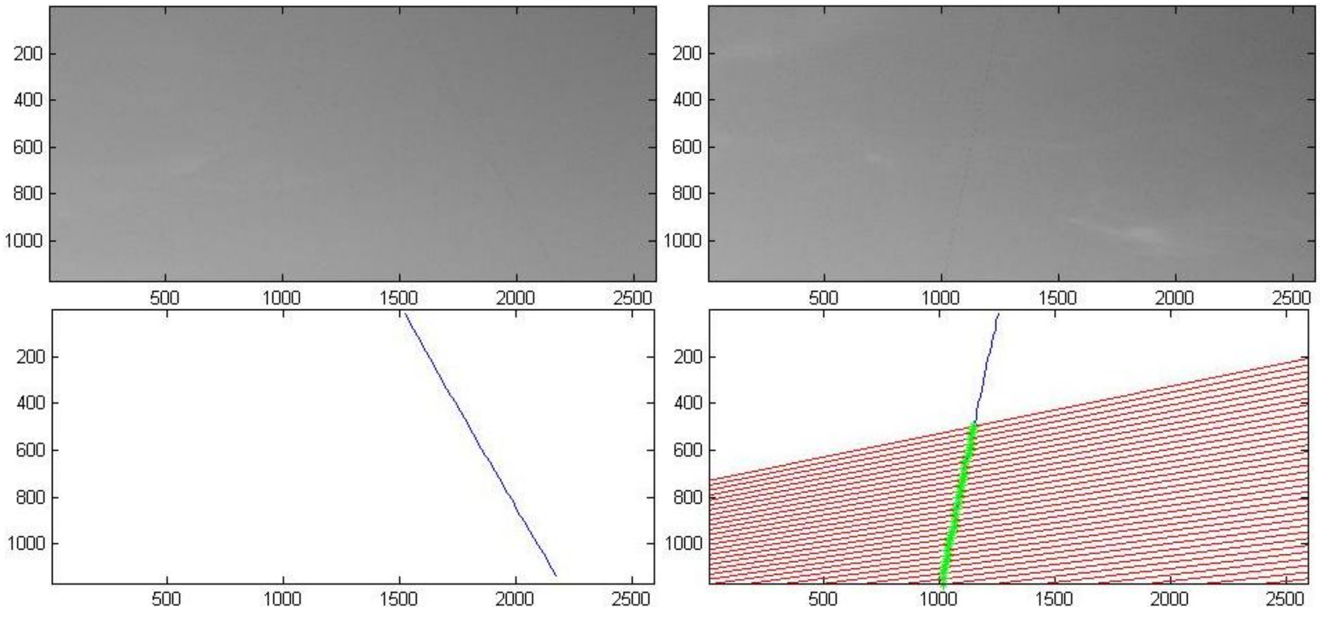

Figure 41. The string is extracted from the left and right images, then corresponding points are found by finding the intersections of the string with the epipolar lines in the right image corresponding to data points in the left image.

The 3D reconstruction of the region of the kite string that was observed for a single frame of an image sequence is shown in Fig. 42.

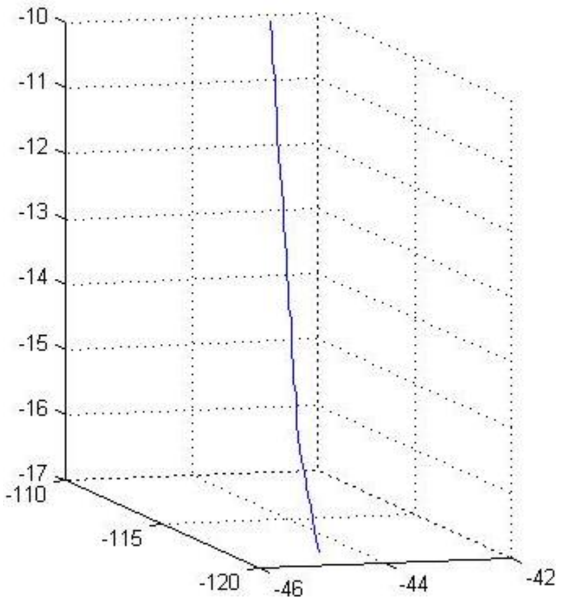

Figure 42. 3D plot of the kite string for a single frame in an image sequence. The scale is in metres.

5. Conclusion

Overall, a good amount of progress has been made, but the system still needs some work and several concerns need to be addressed. The tether detection and computer-vision techniques for reconstructing the tether in 3D that are discussed have been shown to be very effective in a controlled lab environment but can prove to be more difficult to implement in an outdoor setting.

The line detection methods described have been very effective in detecting the position of the tether in an image. Even in images where the tether or kite string is very faint and difficult to see against the background, the detection algorithms have performed fairly well. In fact, it was initially thought that many very high resolution cameras would be needed in order to be able to detect the tether, but it seems that it is actually possible to detect the tether (or at least enough points on the tether) with much lower resolution.

The performance of the tether detection software can be quite temperamental. When it does work, it can work very well and find the tether accurately, but other times it can fail to find the tether at all. This is because the software may not be written with a general enough approach i.e. it is not very good at handling variations between different images.

Furthermore, if various tether detection constraints have been applied to the image as discussed previously (for example assuming that the position and orientation of the tether does not change significantly from one point on the tether to the next), then if the image breaks one of these constraints, the software can fail to find the tether and cannot recover. Therefore, the robustness of the tether detection software is something that needs to be improved.

Through lab experiments, cameras in a stereo configuration have been able to be calibrated using a calibration object. It has been shown that it is possible to reconstruct the 3D position of a hanging cable in the lab using calibrated stereo cameras over a sequence of frames. The same method has also enabled the 3D position of the kite string to be obtained. Calibration of cameras in the field remains a difficult challenge. It was lucky that the dimensions of the Moller Centre building were available to provide a rough calibration, but this would not always be the case. Other ways of calibration need to be explored.

Outdoor experiments using kites highlighted some of the limitations of the system as it currently stands. The variations in lighting conditions can cause problems to the tether detection, i.e. if it is a sunny or overcast day, or if the position of the sun obscures the tether. Another problem was the tether moving out of the range of view of the cameras. This was dependent on the strength and direction of the wind, and the stability of the kite. It may be better to use wider-angle lenses to cover a greater horizontal distance to ensure the tether stays within the frame.

Although the system is able to plot the position of the tether in 3D, it is currently unable to track specific points over time. There is no way of identifying the same point on the tether from one frame to the next. Therefore the displacement of a point on the tether cannot be plotted, so the velocity and acceleration of points cannot be found, at least with the system in its current state. Markings on the tether may have to be employed.

The system was also initially intended to process the data in real time, or as close to real time as possible, using ‘on-site’ computers to extract the data streaming immediately from the cameras. Whether this is actually achievable remains to be seen.

The ultimate goal is to have a system that will enable dynamic analysis of the balloon-tether, ready in time for the full-scale launch, if it is to go ahead, so that the computer models can be validated. The system may have other applications outside of the SPICE project, such as for monitoring the vibrations of cables in structures.

Future work

Further experimental testing is need. To date, the test images and footage has been very controlled, for example the motion of a hanging chain or a suspended cable against a plain background. More tests using kites will give a better indication of how the current system performs in the field, and what measures are required to improve it.

Tracking of specific points on the tether needs to be investigated. As mentioned previously, it may be possible to integrate to a certain length along the tether from the fixed base point. Alternatively, markings on the tether may need to be employed. The effectiveness of using white or retro-reflective markings along the length of the black tether is worth exploring.

If the system is not able to detect the tether sufficiently with a sky background, it may be better to peruse a slightly alternative route. One idea is to fly the balloon at night and use retro-reflective paint on the tether, illuminated using spotlights. This may improve the visibility of the tether against the background so that the tether can be more easily detected. This could mean that fewer cameras are needed and they would not have to be zoomed in as far, resulting in a larger image window for the tether to have the freedom to move in.

Once the system is able to detect and extract the tether accurately and robustly, and once specific points on the tether can be tracked over time, the next step is to perform vibration analysis data, i.e. calculating displacement/velocity/acceleration for points along the tether as well as the frequency spectrum. Comparisons to the existing numerical model can then be made in order to gauge the accuracy of the model.

Other aspects that need to be considered include correcting for ‘barrel distortion’ of the camera lens and possible distortion due to atmospheric effects. An efficient method of calibrating multiple cameras in the field needs to be investigated and may prove to be difficult since such large distances are involved. The current system is also far from real-time, so improvements to efficiency and work-flow will be necessary.

Currently, the use of an aerodynamic tether is being investigated. Possible future work could involve analysing the motion of an aerodynamic tether using cameras.

5. Acknowledgements

I would like to thank Dr. Hugh Hunt for supervising me throughout this project and Prof. Nick Kingsbury for his valuable knowledge of image processing for designing the tether detection software. I would like to thank Hilary Costello and Kirsty Kuo for their help throughout the project, for providing useful resources, advice and assistance. I would also like to thank Prof. Roberto Cipolla for his invaluable lecture course on computer vision.

6. References

[1] Geoengineering the climate: science, governance and uncertainly, [Online.] http://royalsociety.org/policy/publications/2009/geoengineering-climate/ (2009).

[2] Geoengineering by solar radiation management, [Online.] http://www2.eng.cam.ac.uk/~hemh/climate/Geoengineering_RoySoc.htm (2010).

[3] The SPICE Project, [Online.] http://www2.eng.cam.ac.uk/~hemh/SPICE/SPICE.htm.

[4] Wahbeh, A. M., Caffrey, J. P. and Masri, S. F., A vision-based approach for the direct measurement of displacements in vibrating systems, (2003).

[5] Angle of view, [Online.] http://en.wikipedia.org/wiki/Angle_of_view.

[6] The Radon Transform, [Online.] http://cogsys.imm.dtu.dk/staff/ptoft/Radon/Radon.html (1997), Peter Toft.

[7] Firing multiple cameras at the same time, *Breeze Systems*, [Online.] http://www.breezesys.com/MultiCamera/release.htm (2010).

[8] Cipolla, R. and Giblin, P., *Visual Motion of Curves and Surfaces*, Cambridge University Press, (2000).

[9] The Hanging Cable, [Online.] http://webpages.ursinus.edu/lriley/courses/p212/lectures/node14.html (2009).

[10] Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, Cambridge University Press, (2004).

[11] American Society of Photogrammetry, *The Manual of Photogrammetry*, 4th edition (1980).

[12] Frenkiel, F. N and Katz, I., Studies of small-scale turbulent diffusion in the atmosphere, *Johns Hopkins Applied Physics Laboratory* (1956).