

Work Done:**1. Created a list of requirements of the system**

- a. The goal of the data processing is to determine the rate of expansion or compression of the can
- b. Resource allocation
 - i. Resources required for analysis
 - ii. Time to compute analysis (must fit within sampling rate)
 - iii. Space requirements for storing data
- c. Needs to be able to pass data to the client (possibly including processed images)

2. Investigated the use of OpenCV for image processing

- a. Able to use a variety of languages (such as C/C++ and python) giving options depending on what works best with the system
- b. In built functions such as the Canny edge detector, which will allow us to reduce the image down to just the edges, thus making it easier to find the edges of the can to determine its change in size and shape. The canny edge algorithm uses a series of steps to determine what in the image is an edge:
 - i. Use a Gaussian filter to blur the image, such as one of kernel size 5 given by:

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

- ii. Horizontal, vertical and diagonal edges are found in the blurred image by using filters. The first derivative of the horizontal $G(x)$ and vertical directions $G(y)$ are found (by the filters given below), then the edge gradient and direction is found by the hypotenuse and angle between the two

$$G(x) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G(y) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- iii. Non maximum suppression is applied, which removes some of the non-edge pixels by comparing the magnitude of each pixels gradient with nearby pixels, dependent on the gradient angle (i.e. if the gradient angle is 0, then the magnitude must be greater than those east/west of its position)
- iv. Then uses an upper/lower threshold to reduce the pixels further. It is recommended to use a ratio between 2:1 and 3:1
 1. If pixel gradient > upper threshold pixel is accepted as an edge
 2. If lower, pixel is rejected
 3. If the pixel is in between, the pixel is accepted if it is connected to a pixel above the upper threshold

TODO:

1. **Communicate with sensor (team) and sensor interfacing members**

- a. Need to know what file type the image will be and any restrictions it may have
2. **Possibly look into alternate image processing packages and familiarise myself with OpenCV**
3. **Determine a process to calculate the rate of expansion or compression of the can**