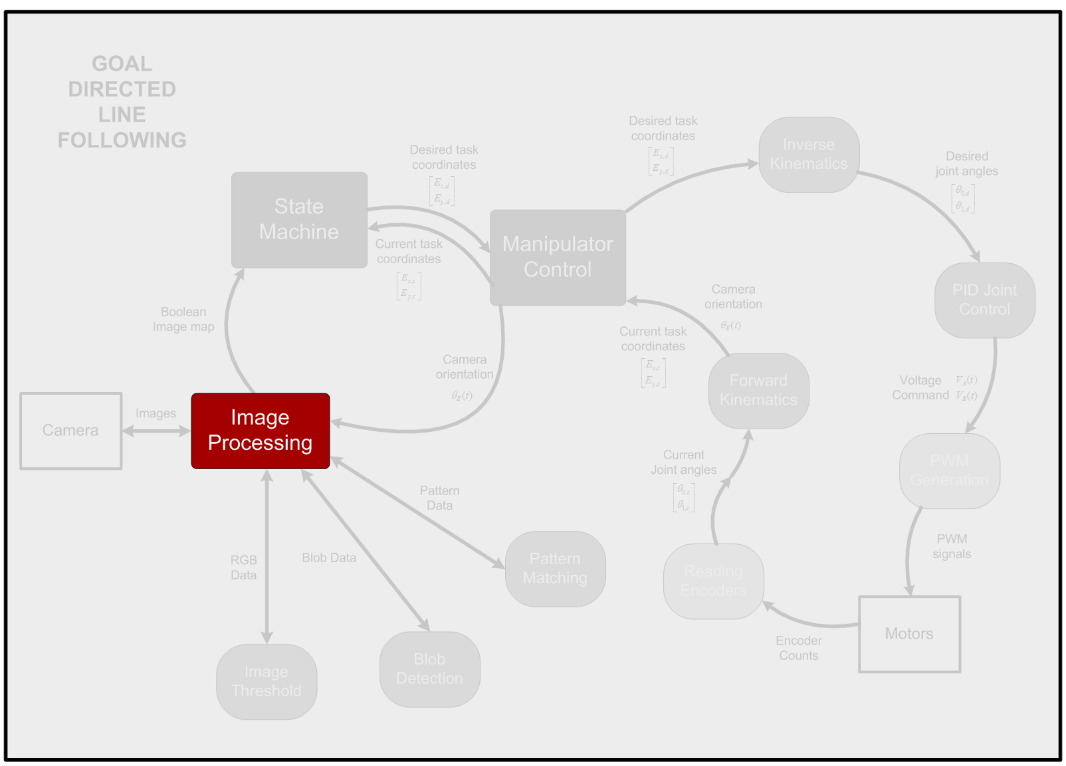
Image Processing

Topics Covered

- Image processing using multiple techniques.
- Image processing with the QNET Mechatronic Systems.

Prerequisites

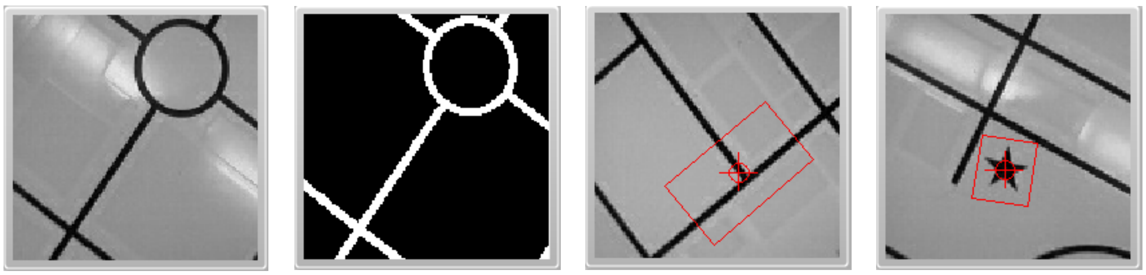
- The QNET Mechatronic Systems is set up according to the Quick Start Guide.
- Image Threshold laboratory experiment
- Pattern Matching laboratory experiment
- Blob Detection laboratory experiment



1 Background

1.1 Image Processing Algorithms

Image processing involves using a series of mathematical operations and algorithms to extract information from a given image. The input is usually a 2-D image, and the output depends on the algorithm used. The process usually starts with thresholding an image to categorize information based on what is required. For example, binary thresholding can be used to cluster all road information as white pixels, and all other information as black as shown in Figure 1.1a and b. Along with algorithms such as pattern matching and blob analysis, the user can identify certain particles from the cluster of road information, such as traffic lights, branches (Figure 1.1c), intersections, specific patterns (Figure 1.1d) etc.



(a) Original (b) Thresholded (c) Branch pattern (d) Star pattern

Figure 1.1: Image processing tools used to extract information from images

For more information on these tools, refer to the Image Threshold, Pattern Matching and Blob Detection laboratory experiments.

1.2 Color and Grayscale Images

Images are represented by 2-D arrays or a function of 2 variables, $f(x, y)$, where x and y refer to the pixel row and column. At each location, the function outputs one or more binary numbers. When a single binary number is provided, the image is considered a gray-scale. For example, using an 8-bit binary number provides a 256 values between black (0) and white (255). Instead, if 3 binary numbers are provided at each pixel location, these can be used to represent a colour image. A common format used is RGB, where the numbers represent the range of colours for red, green and blue. Using 3 8-bit numbers for RGB can provide 256^3, or 16777216 colors, as seen in Figure 1.2.

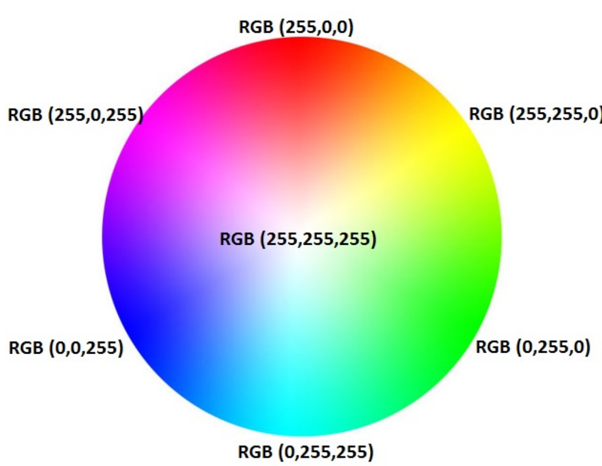


Figure 1.2: 256 color RGB values

2 Image Processing Challenge

In this lab, the QNET Mechatronic Systems manipulator (the *car*) moves along a box trajectory highlighted in Figure 2.1. Along this trajectory, it would encounter 3 traffic lights as well as the plus marker (the *hospital*). The following functionality is desired,

- The *car* moves along the trajectory at a normal speed at all times.
- The *car* must stop when the traffic light is red and proceed when it is green, but may run a yellow light.
- The *car* must slow down when near the *hospital*.

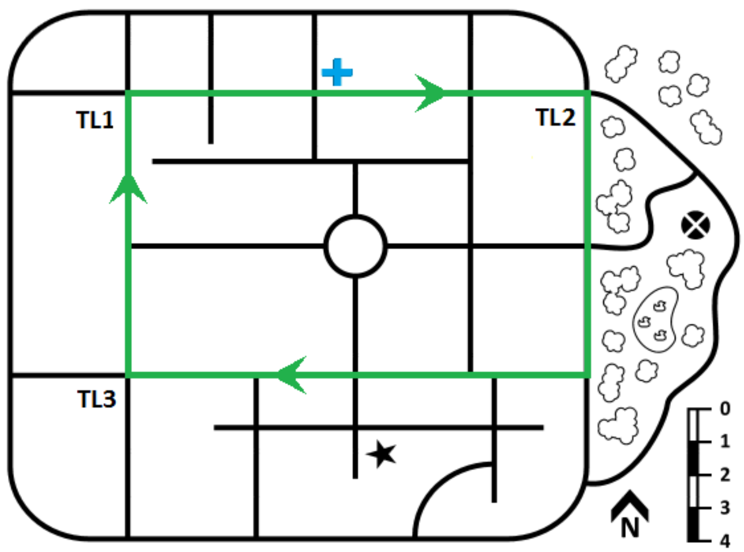


Figure 2.1: Box trajectory along the QNET Mechatronic Systems

The Image Processing loop must provide the state machine with information on whether or not a red light or a plus marker are detected. The state machine and manipulator control loop have already been provided, and take care of the motion delay whenever a stop or slow is required. For more information on these, refer to the State Machines and Manipulator Control laboratory experiments.

2.1 Stage 1: Detecting a Red Light

The first step involves detecting a red light in the image using colour thresholding and blob analysis. Note that although blob analysis will detect the presence of a red light, the *car* should stop only if it is located before the traffic light along the trajectory.

Note: For this lab, ensure that the QNET Mechatronic Systems board is placed in a bright environment. If lighting conditions are poor, the QNET Mechatronic Systems camera's sensors automatically reduce their upper detection threshold to improve their sensitivity in the dark environment. However, this leads to all *bright* objects such as LEDs appearing white. This is called *white saturation*.

1. What RGB pixel values do you expect to see for the red, yellow and green traffic light LEDs in the image captured by the QNET Mechatronic Systems camera?
2. Open `Mechatronic Systems.lvproj`, and under `Quanser ELVIS RIO | Subsystems`, open `Image Processing.vi`. Set Enable Motion? to OFF. Run the VI. After the Calibration bar is full, move the manipulator

manually till the red traffic light LED is at the centre of the image. Move your mouse cursor over the lit red LED in the image, and note the range of the pixel colours in the information box below the Source image. Repeat this with the yellow and green lit LEDs. What values should you set in Red Range, Green Range and Blue Range to isolate the lit LEDs individually? Verify that the Thresholded image shows a cluster of particles accordingly. Do the values match your expectations?

Note: The pixels are ordered as red, green and blue (RGB).

Note: If your LEDs all still appear white, try using `Pulse Generator.vi` that you developed in the Pulse Width Modulation laboratory experiment to dim the LEDs. This change can be incorporated in the Manipulator Control loop.

3. Blob analysis detects the presence of a particle in the thresholded image. Given the functionality required, the *car* must stop for a red light alone, and continue moving if the traffic light LED is green or yellow. Thus, the state machine simply requires a signal which is true when a red light is detected. What RGB range values should you use to make this possible? Verify that with the RGB range you set, only the corresponding LED is detected, and not the others.

4. Connect the output of the Not equal to 0? block after the Blob Analysis block, to the Red Light? block as shown in Figure 2.2, and run the VI with the Enable motion? still set to OFF.

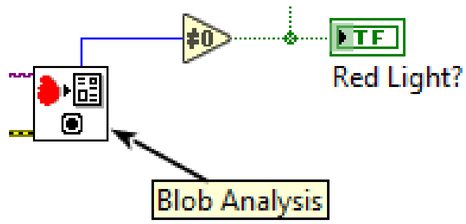


Figure 2.2: Block diagram to indicate whether or not a red light has been detected.

2.2 Stage 2: Detecting the Plus Marker

The second step involves detecting the presence of the hospital (plus marker) in the Source Image, using gray-scale thresholding and pattern matching. The *car* should slow down if in the hospital zone. Recall from the laboratory experiment, that the plus marker was in gray-scale. Comparing a gray-scale marker to a colour image won't provide matching results.

In this section, a different type of thresholding is used. Instead of an RGB format, an HSL format can be used (Hue, Saturation and Luminosity). The advantage of using this notation, is that the luminosity plane by itself refers to the gray-scale version of an image. The RGB colour image is converted to an HSL format, and the Luminance plane is extracted as shown in Figure 2.3. Pattern matching algorithms can be used on this gray-scale image.

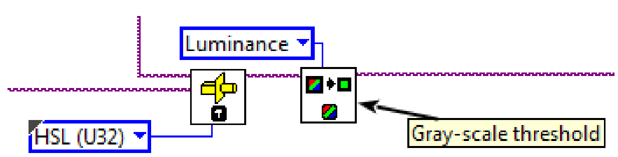


Figure 2.3: Block diagram with gray-scale thresholding.

1. Run the VI with Enable motion? turned OFF. Move the manipulator around and explain any incorrect behaviour? Based on your knowledge from the Pattern Matching laboratory experiment, select Pattern matching settings that enable this correct detection.

2. Connect the output of the Not equal to 0? block after the Pattern Matching block to the Hospital? block as shown in Figure 2.2, and run the VI with the Enable motion? turned ON. Does the VI meet all the functionalities?

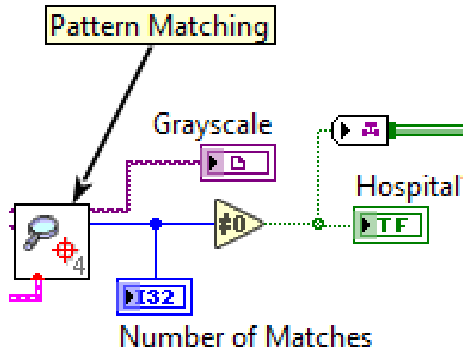


Figure 2.4: Block diagram to indicate whether or not the plus marker has been detected.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com