

CSE 398-011  
Exercise 4:  
Lab Date:

Robocup  
Generating an Edge-based Range Map  
27 Feb 06



- A. **Objectives:** This exercise builds upon the results from two previous labs.
- Using your edge detection code, you will develop a rangemapper that will calculate the position and distance to ground plane objects in the image.
  - Once the rangemap is created, you will use this as input for your IR based obstacle avoider to perform edge-based obstacle avoidance.



Figure 1: Original (l), edge (c) and rangemap images.

B. **The Range-mapper:**

- Your range-mapper function should be defined as follows:

```
void RangeMapper( Image *pEdgeImage, int *pRangeMap, int dX );
```

- There parameter  $dX$  specifies the discretization in the x-direction for the search rays. This parameter should be read in from your configuration file. Note that the size of the rangemap will be a function of  $dX$
- For our initial rangemap, we will assume that there is no roll in the camera (only pan and tilt). As such, the search rays will be along columns. This approximation may be sufficient for the Aibos, as we do not expect *significant* camera roll.
- For the rangemap, edge pixels are obstacles. This is of course not correct, and assumes that all of the background edges on the pitch can be eliminated. This may be true for the grass, but later on we will need to handle lines as a special case.

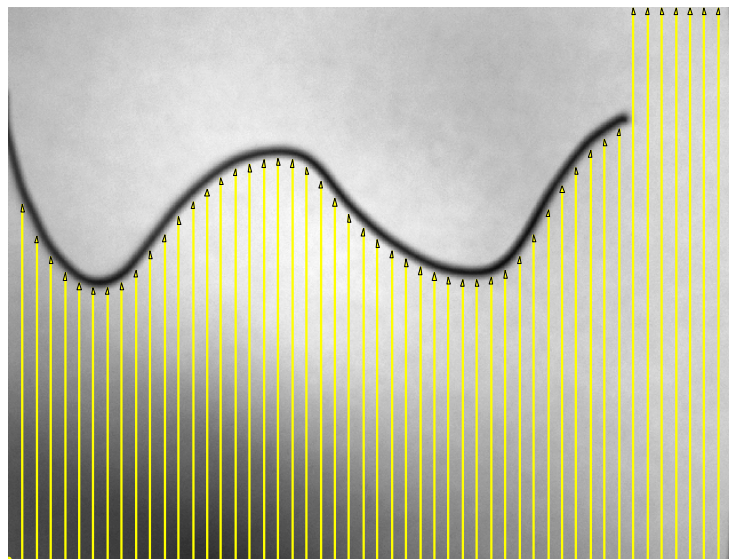
- e. You *must* use pointer arithmetic when stepping through the image pixels. For each column, this amounts to subtracting the number of pixels corresponding to the image width. You *must* keep this as parameter so that your code can be readily ported to the Aibo (or any other platform for that matter).
- f. You can use the drawing functions of the image class to verify that your implementation is working correctly. See

[http://vader.cse.lehigh.edu/docs/libvader/classlibvader\\_1\\_1\\_image.html](http://vader.cse.lehigh.edu/docs/libvader/classlibvader_1_1_image.html)

for more details.

### C. Some Potentially Helpful Tips:

- a. It will probably be a good idea to use something that is easy to segment to test your rangemapper, like the image of black curve on the white paper below:



- b. There may be edge effects on the image, where the border pixels take on exceptional values due to maintaining the convolved image in the same sized image. If you find that you are always detecting the last row in the image, this is most likely the culprit.