Motion Tracking in Aldebaran Nao

Harshal Agrawal
Sarvesh Kumar Singh

Our Target

- With Aldebaran Nao we are trying to achieve following tasks,
  - learn motion flow detection by observing the Informational relations between sensors and actuators.
  - Based on the above learning, perform simple motion tracking based on the motion flow.

Our Work

- Our code is in C and we are using the OpenCV library in this work.
- We are taking the images from Nao and the all the processing of images is being done on the external computer.
- Based on the results of the above processing we are able to find the direction of motion.
- Once the direction of the flow is known, Nao will be able to track that motion by moving its head in that direction.

Algorithm

- 1. We are taking the images from Nao.
- 2. We are reducing these 640x480 size RGB images to 16x16 size.
- 2. We are doing the sensory adaptation (compression of sensory data) on this reduced 16x16 size image.
- The method used by us for data compression is Uniform Binning.

Shrinking the Image

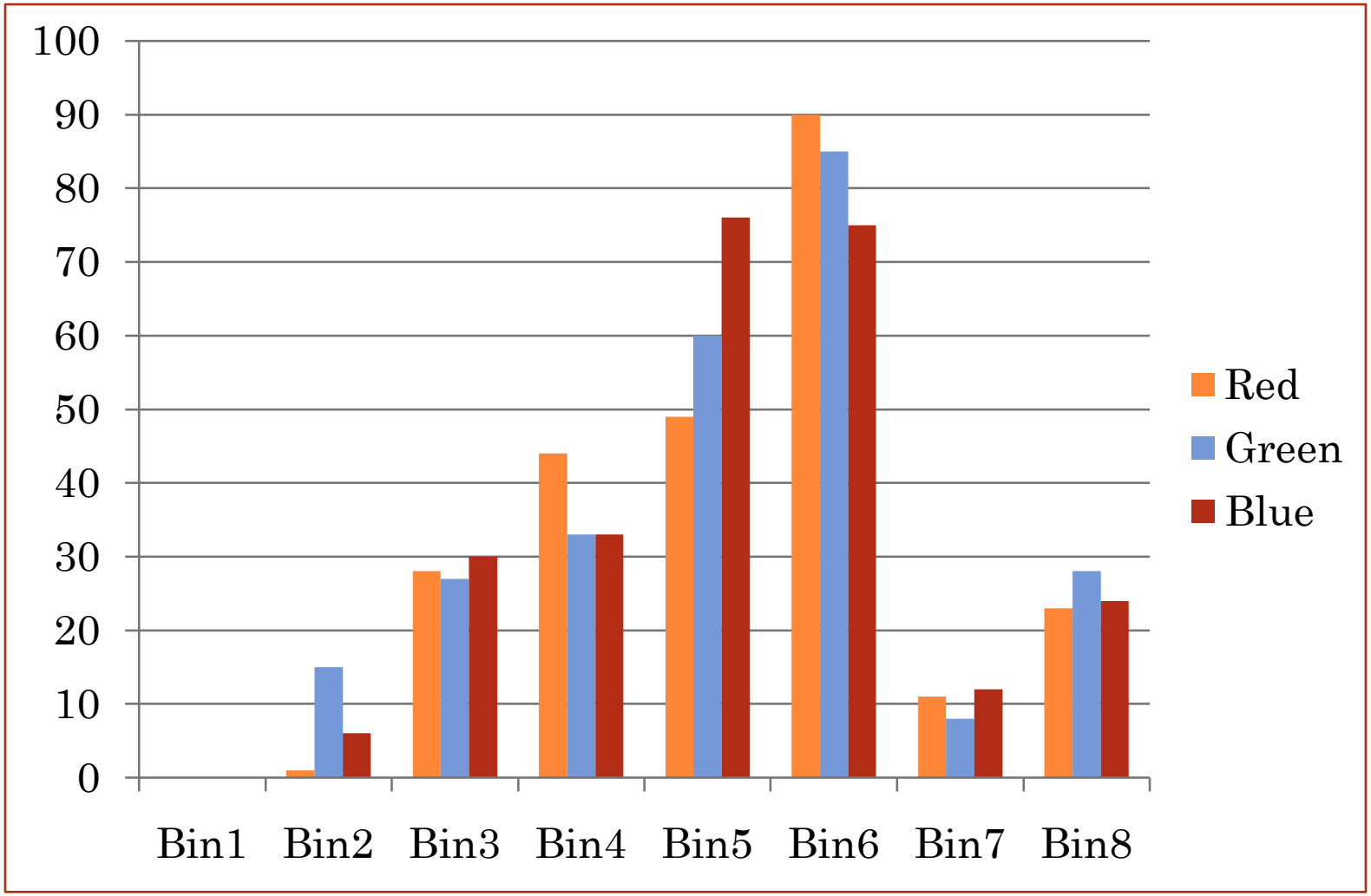
- Images from Nao's camera are 640x480.
- We are reducing these images to 16x16.
- The type of shrinking we require in this work is achieved by averaging the values of pixels in a block of size 40x30 and storing the average value as the reduced pixel.
- In our code this reducing is implemented in the function– IplImage* shrinking2(IplImage* img, int w1, int h1).
- For our case w1 = 40, h1 = 30.

Uniform Binning

- Binning means classifying the data into some classes (ranges of values) to reduce the possible values.
- Now the possible pixel values are 0-255. We have divided the possible values into 8 bins. Bin1(0-31), Bin2(32-63) … and so on.
- This type of binning is called Uniform Binning.
- So now all the 256 pixels of the reduced image are assigned into one of the bins.
- Then we calculated the number of pixels falling into each bin category.

- In our code the binning has been implement ed in the function -- IplImage* bining (IplImage* img).
- This function returns a image in which the pixel values have been replaced by correspoding bin values. Example – 0 for bin 1, 32 for bin2, 64 for bin3, and so on.

0, 1, 38, 44 ,49, 90, 11 , 23



Algorithm ctd..

- Here we have taken 3 sets. Each set consists of 2 images.
- Now we are comparing the center pixel of first image with the all other pixels of second image.
- Then we are counting the number of pixels/sensors having the same value as the center pixel in all three images in all the four directions – up, down, left and right.
- After this we are concluding that the direction having the maximum number of the similar pixels id the direction of motion flow.

Work to be Done

- To improve our algorithm we are going make following changes in our code.
- 1. we will take only two images.
- 2. we will break these images into 4 and 8 parts.
- 3. then we will do the shrinking and binning of these parts too similar to the original images.
- Then we will conclude the motion flow for the pairs of these part images.
- 4. Finally we will conclude the motion flow by finding the direction which is suggested by the maximum pairs of these part images.