

Assignment 4: Color Segmentation in Omnidirectional Images

CS3630 Spring 2007

March 7, 2007

Introduction

In this assignment you will implement a sensor that could be used to localize a robot in a house or other environment: color segmentation in omnidirectional images. Catadioptric omnidirectional cameras provide a 360 degree field of view around a robot by integrating a mirror and a camera. If we can detect colored "landmarks" in these images, this provides us an easy way to localize the robot. Of course, to do it properly, we would need to use probabilistic reasoning on top of that, but that will be the topic of a future assignment.

HSV Color Space

Below, you will be asked to perform color segmentation in the **HSV** color space. The HSV (Hue, Saturation, Value) model, also known as HSB (Hue, Saturation, Brightness), defines a color space in terms of three constituent components:

1. **Hue**, the color type (such as red, blue, or yellow): Ranges from 0-360 (but normalized to 0-100% in some applications) .
2. **Saturation**, the "vibrancy" of the color: Ranges from 0-100% o Also sometimes called the "purity" by analogy to the colorimetric quantities excitation purity and colorimetric purity. The lower the saturation of a color, the more "grayness" is present and the more faded the color will appear, thus useful to define desaturation as the qualitative inverse of saturation.
3. **Value**, the brightness of the color: Ranges from 0-100%.

Color Classification

Color segmentation aims at classifying each pixel in an image into one of a discrete number of color classes. It is an important first step in many color vision tasks. One widely used approach for color segmentation is to use a set of constant thresholds defining a color class as a rectangular block in the color space. For example, in HSV color space, you could sample certain amount of pixels in some interested color region in an image and estimate the mean μ and *standard deviation* σ of these pixels in H, S and V channels respectively. In this way, you could construct your own classifier for this particular color in the following way:

$$|x_i - \mu_i| < \lambda\sigma_i$$

The subscript i represents H, S and V respectively. The λ is some constant value, say 2 for example. If the pixel value x satisfies the formula above, it is classified into this particular color class.

1 Implementing Color Segmentation

1.1 Learning the Color Models

In this section, you need to learn a color segmentation model for landmarks in indoor omnidirectional images.

You can obtain the code distribution and the image data set from

<http://borg.cc.gatech.edu/ipr/files/OT.zip>

Uncompress 'OT.zip' and you will find the omnidirectional images under the directory 'OT/data/'. This image sequence shows that the robot (with omnidirectional camera) goes through living room, kitchen and bedroom. In each room, there are two objects with distinguished color which will be used as landmarks. The landmarks in different room for color segmentation is indicated using a circle in Figure 1.

Please do the following:

1. use your mouse to click on these images to get 10-20 pixels' value for each landmark. Consult the processing reference on how to trap mouse clicks
2. Some code to compute the mean and standard deviation of the color for each landmark in H, S and V channel respectively.

Deliverable

Report on the values obtained for each landmark category in the write-up (see section 1.4).

1.2 Implement Color Segmentation

Please implement color segmentation, filling in the code in 'OT.pde'. We think the following steps will get you there:

1. For any image you want to process, you first need to convert it into HSV color space. The code for this converting procedure is provided in 'OT.pde'.
2. You have to implement a "classifier" function that checks whether a certain color belongs to a certain landmark, using the values from section 1.1. to define the "HSV box" in color space.
3. Then, you apply your color classifiers on every pixel to determine whether it belongs one of those landmarks.

Deliverable

Working Processing code, and an example image with its segmentation in the write-up.

1.3 Plot a "Landmark Graph"

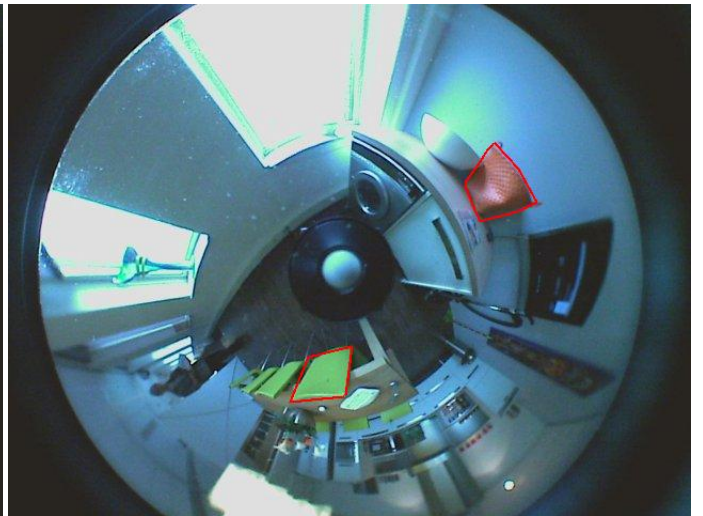
Finally, please apply the segmentation algorithm on *each* image in the image sequence, and plot the classification results on the following graph:

1. the x-axis represents the time (image index)
2. the y-axis represents how many pixels are classified as a certain landmark in a certain image.

Figure 1: Landmarks in Different Rooms



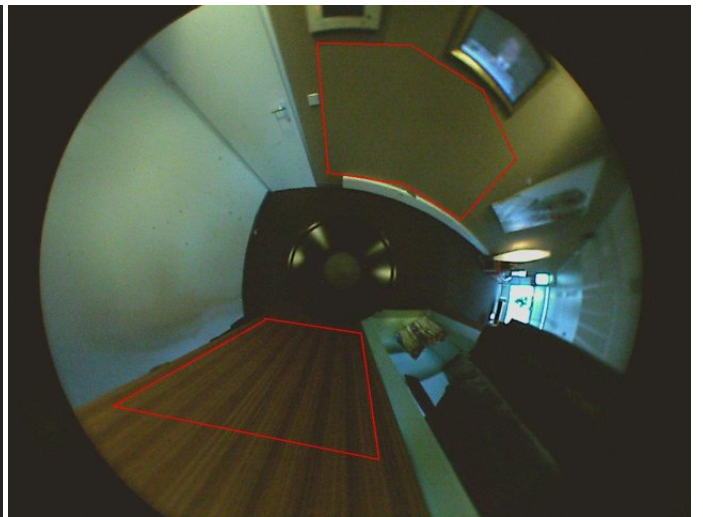
(a) Living room



(b) Kitchen



(c) Bedroom



(d) Hall way

You should plot all of those curves which corresponding to different landmarks on a same graph. For example, you count the amount of pixels which are classified as the green landmark in the living room on every image in the image sequence. Then you plot the dots (i, C_i) with i representing the index of i -th image and C_i representing the amount of the green landmark pixels on i -th image. For other landmarks, repeat the same procedure and use the different colors to draw the curves so that they can be distinguished.

Deliverable

Include in the write-up a picture of the graph and a discuss about what information this graph can provide for the localisation.

1.4 Write Up

The 3 deliverables above together constitute the major part of the write-up, but please also provide some other information on your implementation or anything you want to tell us.