CS231A – Project Milestone Report

Understanding scenes and objects using a set of imprecise cues

Rehan Hameed

Introduction

The main idea of the project is to get inspiration from how humans use a variety of inputs in the process of understanding a scene and the objects in it. Specifically we know that even without the aid of stereovision, humans can do a pretty good job at understanding the depth of scene using even a single eye. And of course that is because over a period of time we develop an understanding about the world and the objects in it. For example human brain can most likely use following types of information in processing a scene:

1. An intuitive understanding of how high the person (i.e. "the camera") is and where can various types of objects lie in the scene with respect to this height
2. An intuitive understanding of the field of view captured by the eyes (i.e how wide or narrow is the view)
3. Are we looking up or down and by what extent
4. Are the eyes focusing at a close distance or at a far distance
5. What properties do objects in the world have e.g.
    a. A car is typically 4-5 feet high
    b. A building can be 5-500 meter high (just throwing arbitrary numbers here)
    c. Sky is close to infinity and is typically blue or gray

So the idea is to have a system which tries to use a similar set of inputs to understand the characteristics of a scene and the objects in it. The general scheme would be to use the approximate information about viewer orientation etc to setup some constraints on scene structure and then use information about the objects including size and features to learn if the scene contains some of the known objects and at what position. Specifically for this study I intend to have a small database of objects (4-5), which the system has "learnt" about.

The first goal was to understand how to measure these parameters in a system and what constraints can be derived from these parameters to help us in the task of understanding the scene and identifying objects. Rest of the document talks about the progress that was made towards this goal.

1. Height

For this study we will assume the viewpoint of a person standing on the ground and this parameter will be set to 5.5 ft. We don't want to rely on knowing exact values of the parameters and each parameter is only going to set loose constraints so this rough value should be fine. If time permits I will try to study the sensitivity of the results with respect to this and other parameters.

Knowing this parameter will give us some information about relationship between image pixels and the world. For example lets assume we know that we are looking dead ahead with exactly a 0 degree vertical angle and we see the following scene:

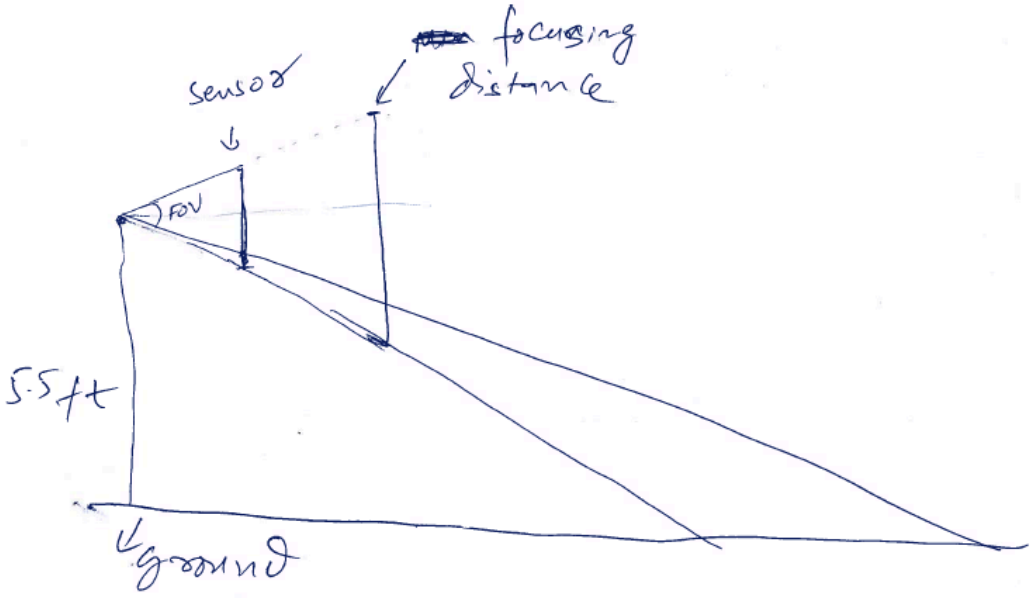
If we are looking dead ahead than the horizon line is exactly in the middle of the image and we know that all the pixels below the horizon line will correspond to a height of less than 5.5 ft and all pixels above it will correspond to a height of above 5.5 ft. And so all the "cars" can only exist in bottom half of the image since according to our database they are less than or equal to 5 ft high. A building top on the other hand will always be above the horizon line as according to our database a building is always going to be at least 20 ft high. Off course our knowledge of the height is "intuitive" (i.e. approximate) so there is a probability associated with which pixels can possibly belong to a car given an approximate location of horizon line and approximate height of our viewpoint. Furthermore at large distances small changes in pixel height results in large changes in object height so this measure does not remain useful. However at very large distances humans cannot distinguish "small" objects like a car and only likely to recognize "big" objects like mountains and that too without a good estimate of the object size / height, so this is consistent with the limitations faced by human vision.

2. Field of view

I am using a canon s95 camera to take my images and taking all images at a focal length of approximately 6mm. So the FOV is approximately 65 degrees in horizontal

direction and 50 degrees in vertical direction. The FOV changes a bit with focusing distance but again we will not worry about knowing the FOV precisely.

Knowing the FOV allows establishing more precise relationships between pixels and locations in the real world. For example if we assume that we are standing on a flat ground plane then using FOV and height we can find a mapping of the ground plane to the pixels:



That gives us a constraint on the maximum possible scene depth for each pixel row below the horizon line. For example given our particular case of the S95 camera at 6mm and a nominal height of 5.5ft, the lowest pixel of the image row can correspond to a scene depth of 12mm or less. A pixel row at 20% of image height on the other hand can represent a maximum scene depth of around 18m.

Again the constraint becomes more error prone as we get close to the horizon line. However our desired accuracy also goes down as we move to farther and farther distances. For example if a particular part of the scene is thousands of meters away then it does not really matter whether it is 1000m or 10000m.

3. Depth estimate from focus distance

Camera systems utilize either contrast detection or phase detection to determine the distance to get an image area in sharp focus. This step aims to use this mechanism to model the knowledge about whether we are focusing on a near object or a far object. Canon S95 uses the contrast detect autofocus and allows us to move

the focus area to any image region (in steps). It also reports the focusing distance information as part of the image EXIF data:

Using this it is possible to create a 21x21 grid of focus depth values for the whole image. Again the accuracy would be less at greater depths, which is not a problem for our intended use. At this point I am doing this process manually, which has the down side of being a slow process and requires taking a large number of pictures. However I am looking into using Canon Hack Development Kit – a free development kit that allows writing scripts to completely control the operation of most canon cameras. Using that kit it might be possible to have the camera automatically focus on each image region one-by-one and at the end return a coarse depth map of the scene.

That information then works with the rest of the information to give us more clues. For example if an image region is at a large depth – say 1000m then we are not expecting to find an object like say a car which is very small in relation to this depth. As another example by using the depth and FOV information we can have an estimate which regions of the picture likely belong to the ground plane. Plus using the depth and FOV information gives a rough estimate of the "height" of various image regions.

4. Vertical viewing angle

Most of the previous discussion assumed that we are looking straight ahead such that the horizon line is exactly in the middle of the image, with the lower half of the image representing the "ground region" and upper half representing the "sky region". Of course we will rarely be looking precisely straight ahead and often we can be looking up and down. The system will assume that we have some approximate knowledge of this vertical viewing angle, which then gives us an idea on the expected location of the horizon line. For example following picture shows

the calculated range of possible locations for the horizon line based on the assumption that we are looking 10 degrees down and assuming that the vertical viewing angle is known to a resolution of around 5 degrees.

For this study I am considering one of the two mechanisms. First is to either use an "intuitive" measure of how inclined is the camera. Second is to use the digital level gauge found in some recent SLR cameras such as my Sony A580. The electronic level gauge in a580 seems to be accurate enough to give the 5 degrees accuracy, which I am assuming as required at this point. Whereas the "intuitive" method might have a larger error.

There can be a couple of ways to improve the estimate of the horizon line. One would be to use this estimate as input to an algorithm, which tries to find the horizon line by finding lines converging to a vanishing point. The second is to use the depth information from previous step to help fine tune the location of the horizon line.

5. Object information

To keep the task manageable I am assuming that the system knows about only 4 object classes in the world and that too from a roughly a single view point (for example it has only ever looked at a car from behind, not from a side). The data that the system knows would include the range of heights and widths for each object class as well as SIFT-type features characterizing the class. Following object classes will be used:

1. Cars
2. Boxes
3. Buildings
4. Road Signs

There are two ways in which this information about the objects work with the previous sources of information. First, the structure of the scene estimated from the previous sources puts constraints over which parts of the scene can contain which of the objects. For example in the case the above case with a 10 degree downward viewing angle, there is a very small part of the image which can likely have a traffic sign in it. Similar if there is an object which is roughly 10meters away, and based on its pixel locations it is estimated to be 5 meters high then it is not likely to be a box and so on.

Conversely, as successful object identification can provide clues about the scene: “This portion of the image looks like a car and based on its estimated size in pixels it must be at this much depth because cars tend to be this much size in real world”.

The main goal is to come up with a smart algorithm which derives information from all these non-exact sources and gives a relatively refined estimate about the scene including the depth at various points, location of gorund plane if visible and identification and location of any recognized object classes.