CS664 Lecture #4 | Matching | 9/13/01

Scribe: Tom Wexler | Lecturer: Ramin Zabih

1 Introduction

Last time we discussed Least Median Squares ($LMedS$). $LMedS$ handles fitting optimally with respect to outliers, having a breakdown point of $(50 - \epsilon)\%$. We can view $LMedS$ as finding the thinnest ruler which covers at least half of the points given (we assume that the remaining points are potential outliers). However, under some reasonable assumptions about the noise, we can do better. For example, if we assume that the noise is approximately uniformly distributed, $MINPRAN$ will perform better than $LMedS$. In this lecture we consider the problem of matching in two similar problems.

2 Stereo Matching

Given two images related, we would like to establish a visual correspondence between them. In otherwords, we are asking 'what went where?' One basic example of this problem is stereo matching.

Suppose we are trying to use two images of an object viewed from different angles to discern its three dimensional structure. It would be helpful to have a map which told us which pixels in the first image correspond to which pixels in the second. We have the benefit of knowing the positions and angles of the two cameras, and we know that the pictures were taken that the same time.

Unfortunately, we still have a number of problems. For example, consider a point on the surface of the object we are studying. Not only will the pixel most likely be in a different location in the second image, but it also may change in color and brightness. Some materials, such as metal, mirrors, or skin, have an appearance which is highly dependant on the viewing angle. Furthermore, many cameras have independently selected parameters called gain (α) and bias (β). The calculated intensity of a pixel as reported by a camera is

$$I' = \alpha I + \beta$$

where I is the actual observed intensity. If the gain and bias of the two cameras are unknown, this further complicates the task of matching related pixels.

Considering human stereo is not particularly helpful, as we only use stereo at close range. Even then, we use a number of other techniques for estimating depth, such as size, brightness, overlapping, motion, and a basic understanding of the spaces we are considering.

3 Motion

Another similar problem is that of interpreting motion. Here we assume the camera is fixed but our images are temporally separated. For simplicity we can assume that motion is one dimensional. Therefore each pixel has a displacement, which is proportional to $\frac{speed}{depth}$. Our task is to find the best displacement for each pixel.

To simply matters, we assume an upper bound on the speed of any object, effectively bounding the maximum dispacement. In fact, we will consider the case in which we assume each pixel p does one of two things: p remains stationary (moves 0), or p moves one step to the right. If we define $L(p)$ to be the intensity of pixel p in the first image and $R(p)$ to be the intensity of p in the second image, we can define a cost function as

$$c(p, d) = (L(p) - R(d(p)))^2$$

. Here $d()$ is the displacement function, so $d(p)$ is either p or the pixel to the right of p.

4 Problems with Noise and Borders

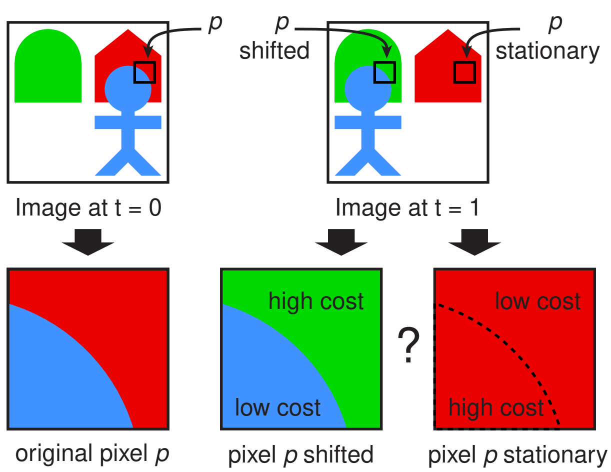

Figure 1: Problems dealing with motion at border pixels.

Of course, as with stereo matching, we will have the problem of noise. One solution is to use spatial averaging. Unfortunately, this approach has a number of problems. First, local averaging gives garbage answers at border locations. Consider Figure 1. Pixel p is shown containing both the colors blue and red. We seek to determine whether or not p is shifted in the second figure. Clearly, we want our algorithm to decide to shift p, but it may not do so. If it does the shift, it benefits from the low cost of matching the blue to the blue, but incurs a high cost for replacing the red with the green. In staying put, our algorithm pays a high price to change the blue to red, but the green stays the same, which implies a low cost.

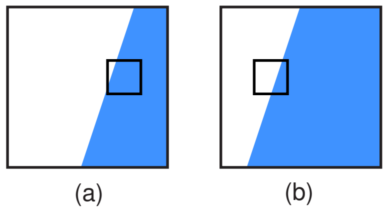

Figure 2: More problems: Sampling errors.

Second, we have sampling errors. Consdier the blue object moving in front of a white background as in Figure 2, and consider a pixel on the border of the object which contains both blue and white. The pixel shown in Figure 2(a) is predominantly blue, while the corresponding pixel in Figure 2(b) is predominantly white. The problem arises because our image is necessarily discretized.

5 Getting Around Problems with Borders

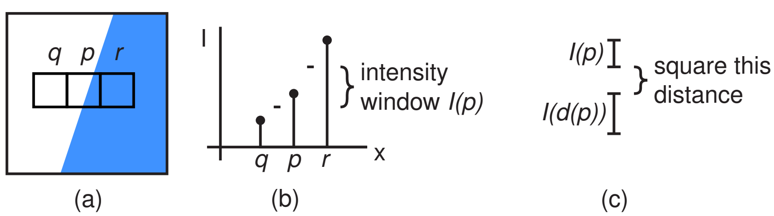

Figure 3: A solution: Intensity windows.

To deal with this sampling problem, Birchfield and Tomasi introduced the method of using intensity windows to acknowledge a lack of precision in areas of rapid change. Instead of considering a single pixel p, we consider the pixel along with its neighbors (Figure 3(a)). We now view the value of pixel p to be somewhere in the range defined by the average of p's value and the value of its neighbors, as shown in Figure 3(b). To calulate the penalty for selecting displacement $d()$, we consider the distance between the windows of each pixel and its image, and then apply the appropriate cost function (such as squaring, possibly). See Figure 3(c).

6 A Global Approach

At this point, we have improved upon our initial and naive approach, but we still do not get great results. We have ignored one important piece of information that could help in solving the matching problem. Namley, we expect that pairs of adjacent pixels will remain adjacent, even if both move to a new location. No such consideration was made with the local approaches used previously: Each pixel decided where it would go independently of where the remaining pixels were sent. This has the advantage of being fast and easy to implement, but often gives poor results. By switching over to a global method, we hope to spend more time in exchange for a better solution.

Global methods can be thought of as optimization problems. We have a space of possible solutions, and an objective function that evaluates the quality of a given solution. We will often consider the objective function as an energy function which we would like to minimize. With that high level view of global methods, we are left with two questions. First, what kind of energy function should we consider, and second, given such a function and a space of solutions (ridiculously large), how do we minimize the energy? These questions will be addressed in the following lectures.