Stereo Vision for Obstacle Detection in Autonomous Navigation

Brendan M. Collins[1]
Alain L. Kornhauser[2]

Princeton University

May 24, 2006

[1]Class of 2008, Computer Science
[2]Professor, Operations Research and Financial Engineering

Contents

List of Figures

List of Algorithms

Acknowledgements

I would like to thank my advisor, Professor Alain Kornhauser, for his unwaivering support and tremendous enthusiasm for this project. His committment to keep this project a student endevour gave the entire team unparalleled academic opportunities. My sincere gratitude also goes to all of my teammates. Without the vision, tanacity, and ingenuity of every team member, this work would be meaningless. I also wish to acknowledge the financial supporters of the project, including the CSX Transportation Research Endowment Fund, Eric Huber, Katherine Kornhauser, General Motors, Applanix Corporation, Tom Haines' "Pick Your Own" Blueberry Farm, and ALK Technologies, Inc.

Introduction

Autonomous vehicles have the potential to revolutionize both civilian and military transportation. In the military sector, using a human driver for non-combat missions needlessly puts a soldier at risk. Indeed, the 2001 Defense Authorization Act requires that by 2015, one third of all military ground vehicles be unmanned. Toward this goal, the Defense Advanced Research Projects Administration (DARPA) announced the 2004 DARPA Grand Challenge - a 130 mile, completely autonomous race across the deserts of California and Nevada. No entrants completed this race, and the furthest entry made it only 7.4 miles. DARPA announced a second competition to be contested in October, 2005. Entrants to the challenge used a wide variety of vehicle platforms, sensing techniques, and control methods. Princeton's entry to the Challenge, Prospect Eleven, was unique in that its only environmental sensor was a stereo camera pair. This paper describes the stereo vision techniques used by Prospect Eleven in the Grand Challenge, as well as subsequent work in the field.

Grand Challenge teams used two types of sensors to avoid obstacles and stay within road boundaries: LIDAR (Light Detection and Ranging), and vision. LIDAR units tend to be very costly, as they have very limited applications. Furthermore, most LIDARs only give one line of the scene at a time, so range measurements must be integrated over time to obtain a complete picture of the terrain ahead. This process requires delicate sensor stabilization, and further increases the cost of using LIDAR. In keeping with the Princeton team philosophy of simplicity, LIDAR was rejected as being too complex and too expensive. The obvious alternative was vision. A monocular system was developed by Michael Pasqual '05, which optical flow and color information to identify obstales. To compute the location of obstacles in the scene, the image was transformed into the orthographic plane, and pixels corresponding to obstacles were located relative to the vehicle's posi-

tion. Conceptually, this system supposes that the ground ahead is a plane parallel to the motion of the car, and everything in the image is painted on the plane. As natural scenes do not always meet this assumption, the system was not very robust at ranging obstacles. Additionally, optical flow is plagued with difficulties, particularly in very homogenous scenes (such as much of the desert), or when the vehicle goes so quickly as to make a pixel location in one image correspond to an entirely different object in the next image. A stereo pair, in which two cameras are mounted a fixed distance apart, was used instead. Though not without its limitations, this technique was adequate to qualify Prospect Eleven for the Grand Challenge, and even scale several mountain passes at night [6].

Section 1 of this paper discusses the technique of Stereo Vision. Section 2 discusses the particular problem of obstacle detection for the Grand Challenge, and proposes a simple algorithm to accomplish the task. Section 3 proposes an algorithm for tracking obstacles in the time domain. Finally, Section 4 discusses work to extend the range of stereo vision, and considers future work in the field.

Chapter 1

Stereo Vision

Stereo vision uses two cameras separated by a known distances to obtain a map of the terrain ahead. In particular, an image is captured from each camera simultaneously. Then, features in one image are matched to features in the other. Finally, using these matches and information about the relative position of the two cameras, the position of each feature relative to the camera is computed. This section provides an overview of this process, drawing heavily from [11].

1.1 General Principles

Figure 1.1 illustrates the governing principle of stereo vision. The point P is some point in the scene. C_l and C_r are the centers of projection for the left and right camera, and I_l and I_r show the cameras' respective image planes. Given only the position of P in the image plane the left camera, P could fall along any point on the ray from C_l to P. But by determining the position of P in the image plane of the right camera, the position of P can be triangulated by determining the point of intersection of the two rays. More formally, let b be the distance between C_l and C_r, p_l and p_r be the location of P in the image plane of the left and right camera, f be the focal length, and Z be the distance from the baseline to P. By similar triangles, we have

$$\frac{b - p_l - p_r}{Z - f} = \frac{b}{Z} \tag{1.1}$$

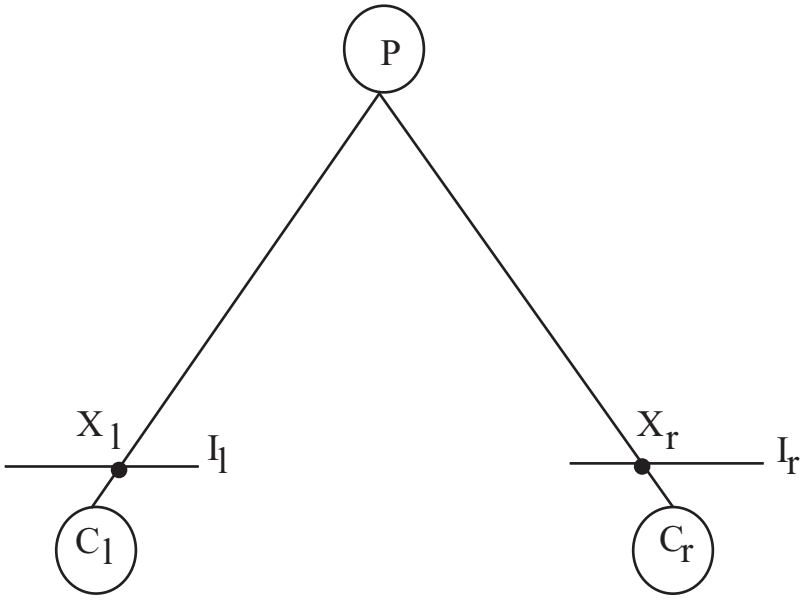

Figure 1.1: Principle of Stereo Vision

This equation gives an expression for Z

$$Z = \frac{fb}{p_r - p_l} \tag{1.2}$$

Except the quantity $p_r - p_l$, all terms are known properties of the system. As such, $p_r - p_l$, termed the *disparity*, is all that must be computed to determine the position of features in the scene. The next section examines an algorithm to compute disparity.

1.2 Disparity computation

If possible, it is desirable to compute this disparity for every pixel in the image. Algorithm 1 accomplishes this. For each pixel in the left image, the algorithm attempts to minimize a matching cost function ϕ over the right image subject to the constraints of camera geometry.

Though many cost functions may be adequate, an efficient and effective one to sum the squared difference over some $2k+1x2k+1$ window. Naturally, the window is centered around the pixels which are being considered for a

Algorithm 1 Stereo Matching Algorithm

1: Let $I(r, c)$ be the pixel value for the pixel in row r, column c of image I
2: Let $D(r, c)$ map each pixel $I_l(r, c)$ to its corresponding feature in I_r
3: Let $\phi(r_l, c_l, r_r, c_r)$ be a function which computes the cost of matching $I_l(r_l, c_l)$ to $I_r(r_r, c_r)$
4: **for** each pixel r_l, c_l in I_l **do**
5: **for** each possible match r_r, c_r, in I_r **do**
6: Compute $\phi(r_l, c_l, r_r, c_r)$
7: **end for**
8: Let $D(r_l, c_l)$ be the choice of r_r, c_r which yields the minimum value of $\phi(r_l, c_l, r_r, c_r)$
9: **end for**

match. Larger windows tend to produce superior results, as more information is available in matching, but it also slows computation. The formula used is

$$\phi(r_l, c_l, r_r, c_r) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} (I_l(r_l + u, c_l + v) - I_r(r_r - u, c_r + v))^2 \qquad (1.3)$$

Significant efficiency gains may be realized by limiting the set of possible matches considered in steps 5–7. Figure 1.2 shows a point P's projection onto the camera planes of the left and right cameras, π_l and π_r. As noted before, given the location of a point P in camera plane of the left camera, its position in the world is limited to the ray $C_l\vec{P}_l$, starting at C_l in the direction of p_l. So to find P in the right image plane, only the projection of $C_l\vec{P}_l$ onto the image plane of the right camera need be considered.

To even further simplify the detection algorithm, right image may be rectified so that each ray from the left fixation point is project into a row in the right image plane. In effect, the algorithm need only search a single row of the right image for each match in the left. Moreover, if it can be assumed that objects will be at a certain minimum range, the number of columns searched can also be capped. This rectification is quite simply in the case of two cameras parallel to one another, as was the case in Prospect Eleven. Indeed, if lenses were ideal, no correction would be necessary at all. However, as a result of radial distortion and subtle imperfections in the lens, slight rectification is performed before matching. Once matching is performed for each pixel, a *disparity map* may be generated. A disparity map, denoted

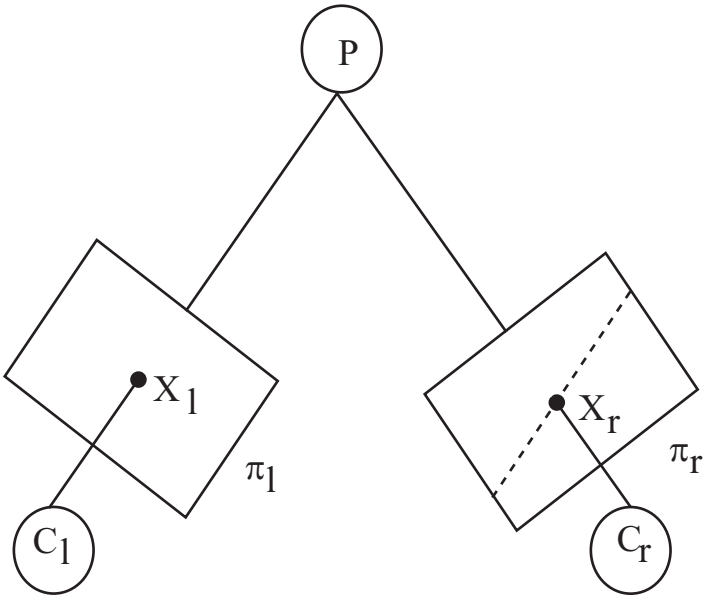

Figure 1.2: Projection of $C_l\vec{P}_l$, the ray from the left center of projection to the image plane, onto the right image plane π_r

$D(r, c)$ in the algorithm, is simply an image in which the intensity value of a pixel is the disparity of that pixel with respect to a stereo pair. Figure 1.3 shows a stereo image pair, and the corresponding disparity map.

1.3 Implementation and Results

In order to maximize development time, a commercial stereo camera was purchased. A Point Grey Research (Vancouver, Canada) Bumblebee captures simultaneous images from two black and white CCDs. The baseline separation is 12 cm. The included libraries implement Algorithm 1. Additionally, the libraries perform validation on the pixels in order to determine if a disparity value is likely to be accurate. In particular, the algorithm considers the amount of variation in a window, and if the variation is very low, the match is declared unreliable. Invalidated pixels are displayed as black in all figures in this paper.

Several strategies were found to be effective in improving the number of accurate and validated matches—particularly in poor lighting conditions.

Figure 1.3: The left image, right image, and disparity map for a natural scene

Red photographic filters mounted in front of each lens were used to increase contrast by blocking blue light and reducing UV haze, mitigating problems such as CCD "bleeding" on bright days and boosting the brightness of the ground, the area of interest in each frame. Results were further improved by custom camera gain control which was designed to optimize the exposure in the ground plane at the expense of the upper half of the frame. Given the current gain G, the average intensity sampled over some region of interest C, and the desired intensity value T, the new gain G' is:

$$G' = G + k(C - T) \tag{1.4}$$

where k is simply a scaling constant. Typically, T is selected to be rather low, in the range of 80-100, and C is calculated by sampling approximately 1% of the pixels in the lower-central portion of the image.

Even under perfect lighting conditions, there remain many limitations to stereo vision. Certainly, range is a limiting factor in many applications. In practice, 20m is the maximum reliable range. Figure 1.4 gives a plot of range at various disparities. At the minimum reliable disparity value of 3, the range is 20m. Figure 1.4 illustrates another serious limitation of stereo vision. As the difference in range can be very large for neighboring disparity values, and disparity falls only on integer values, there is significant error caused by quantization. Even worse, if disparity value is off even by one, a tremendous error in ranging can result. Figure 1.5 illustrates the error caused by quantization, assuming perfect disparity values. It clearly illustrates that error rises drastically as range increases.

Unfortunately, there is nothing to be done about the intrinsic errors of stereo vision but to design around these sources of error. The next chapter presents an obstacle detection algorithm which is well-suited to the errors of

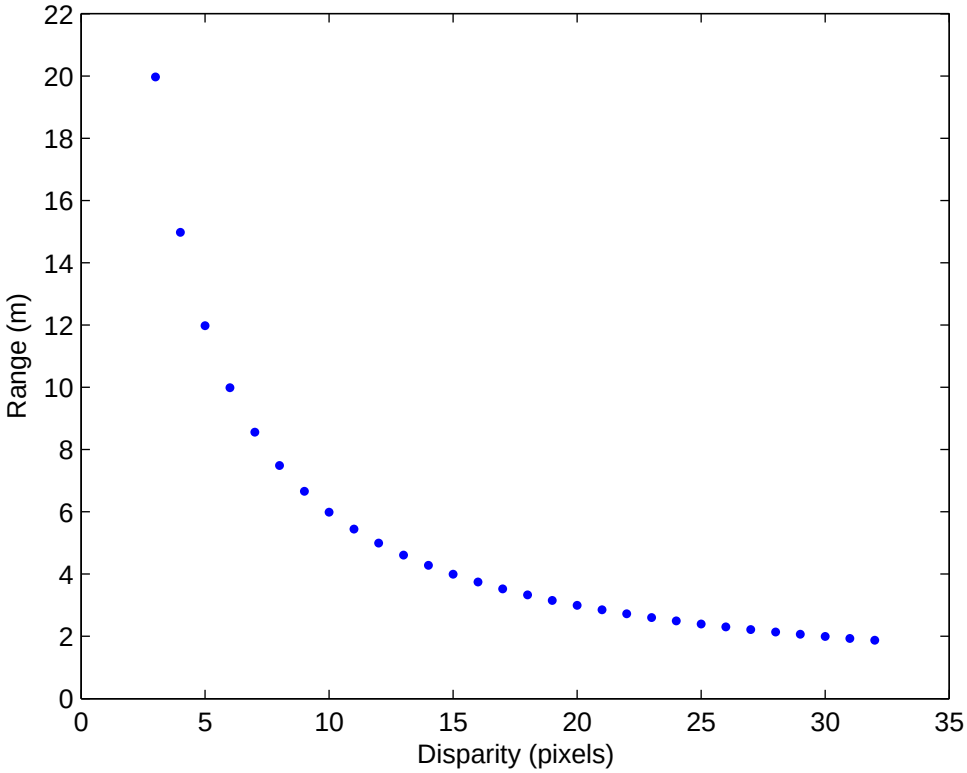

Figure 1.4: Disparity versus Range for the stereo camera used by Prospect Eleven

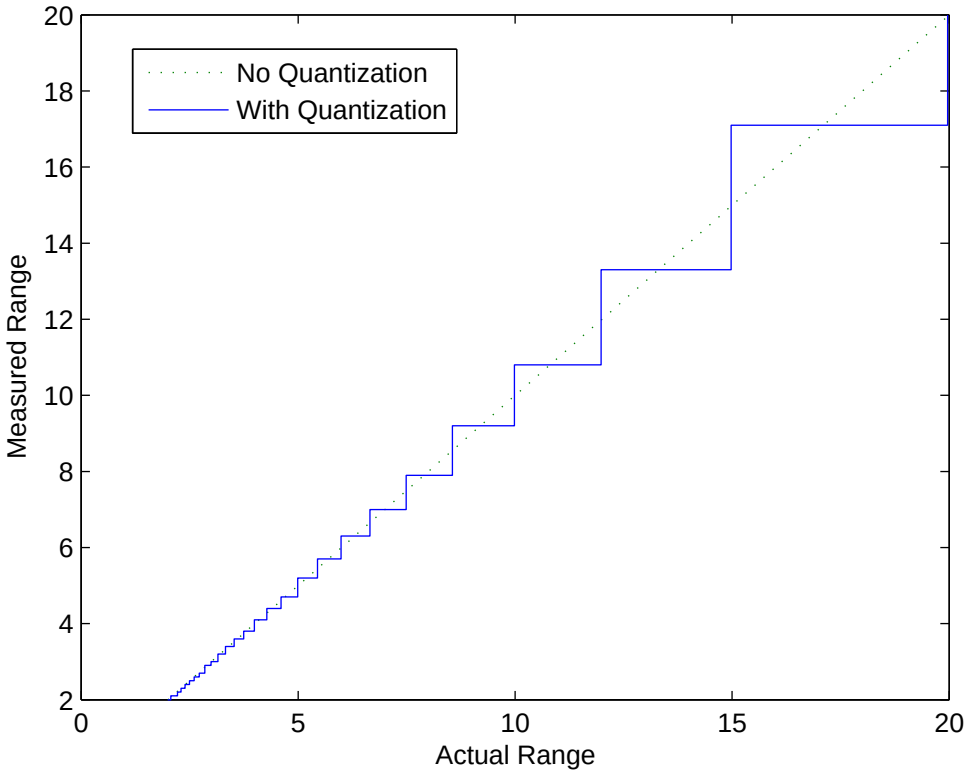

Figure 1.5: Actual and measured range, with impact of quantization shown

stereo vision, and to the Grand Challenge.

Chapter 2

Obstacle Detection

Given the ability to acquire depth maps as a function of several parameters, this chapter presents an algorithm used to detect obstacles in the depth map. The choice of algorithm is informed by the errors intrinsic to stereo vision, as discussed in the previous chapter. In particular, the algorithm presented is designed to be as fast as possible, such that many possible measurements of an obstacle's position may be obtained. These measurements are combined to obtain a superior measure of position, as discussed in the next chapter.

Figure 2.1 gives the coordinate system which shall be used for the remainder of this paper. The origin is at the camera, the heading of the car is the positive y axis, to the right of the camera is the positive x axis, and above the camera is the positive z axis.

2.1 Previous work

Obstacle detection algorithms may be divided into two sets, those that find obstacles in the disparity map (see [2], [4], [5], and those that first transform the disparity map into a point cloud and then work in 3-space (e.g, [13], [10]).

Working on the disparity map allows algorithms to work with data in a form which most closely represents the way it is acquired by the sensor. Moreover, as only 2 dimensions are used, working on the disparity map is typically less expensive in terms of storage and computation. The primary advantage of the latter approach is that it need not make assumptions about the position of the camera relative to a ground plane, whereas most disparity map approaches must. However, as Prospect Eleven has the camera mounted

Figure 2.1: Coordinate System

securely to the hood, it is safe to assume that the ground plane is roughly perpendicular to the camera plane.

The approach of [2] is to simply assume that a ground plane extends out in front of the car, and mark as an obstacle any object which does not fall in that plane. The plane in this case takes the form

$$z = c \tag{2.1}$$

for some constant c.

Similarly, [13] relies on the ground plane assumption, but dynamically computes the ground plane based on the current disparity map. In this case, the ground plane takes the form

$$ax + by + cz + d = 0 \tag{2.2}$$

for some constants, a, b, c, and d.

Other approaches to not look at global features of the image, but instead look only at small regions. For instance, [8] calculates the radial slope of each pixel, and marks as obstacles those pixels which are above some threshold. Similarly, [5] limits the maximum slope relative to the ground plane. Both approaches are advantageous in that each pixel need only be analyzed with respect to its neighbors as opposed to in comparison to the entire image. Simplifying even further, [3] simply searches for sufficiently large regions which are at sufficiently similar depth values. The approach of [3] reduces to

that of [5] and [8] if it is assumed that the ground plane extends in front of the camera, and that the camera plane is perpendicular to it.

The approach of this paper combines the global ground plane height thresholding techniques of [13] and [2] with the slope thresholding techniques of [8], [5] and [3]. The two metrics can be integrated in a simple and elegant manner.

2.2 Finding the ground plane

As the Grand Challenge course was graded, it is reasonable to assume there exists a flat, traversable region in front of the vehicle. However, it is not necessarily the case that this region will be entirely flat. It is both simpler and more accurate to simply assume that in each row of a disparity map, the disparity value will be constant for pixels which correspond to traversable terrain. Expressed mathematically, we have that the height of the ground 'plane' is a function of y.

$$z = f(y) \tag{2.3}$$

Obviously, this implies that

$$\frac{\partial z}{\partial x} = 0 \tag{2.4}$$

which is precisely the constraint suggested.

It now only remains to compute $f(y)$. The method suggested in this paper is to compute the row-wise median in the disparity image. This method assumes that at least 50% of each row is not an obstacle; experimentation revealed the assumption to be reasonable in most cases. The failure modes of this assumption is discussed later in this chapter. An additional advantage of this method is that as a result of the quantized nature of disparity values, the median may be computed in linear time using Algorithm 2, based on `BINSORT,` from [7].

2.3 Implementation and results

Were range data perfect, a logical measure of slope in the scene would to simply measure the rate of change of the disparity values. Moreover, it is

Algorithm 2 Linear-Time Median Calculation

Let $D(r, c)$ be the disparity value of row r, column c in disparity map D
Let $M(r)$ be the median disparity value in row r
for each row r **do**
    Let $Count(i)$ be the number of disparity values in row r which are equal to i
    Let $Count(i) = 0$ for all i
    for each pixel c in row r **do**
        Increment $Count(D(r, c))$
    end for
    Let $DisparityValue = 1$ be the disparity value considered in the current iteration of the following for loop
    Let $Accumulation = 0$ be the number of pixels with value $\leq DisparityValue$
    while $Acculation <$ Half the number of columns in row r **do**
        Add $Count(DisparityValue)$ to $Accumulation$
        Increment $DisparityValue$
    end while
    Let $M(r)$ be $DisparityValue - 1$
end for

assumed the ground plane is perpendicular to the xy axis, so high slope with respect to the ground plane is equivalent to very low slope in disparity over an interval of rows. Additionally, quantization renders measures like variance meaningless. As a result of quantization, regions of high slope with respect to the ground plane appear as long contiguous intervals of rows all at the same disparity value. A natural algorithm, then, is to simply search each column for an interval of contiguous rows at the same disparity value. If the interval is longer than some threshold, l, the span is declared an obstacle. The median ground plane criterion may be incorporated by varying the value of l in according to the extent to which the top pixel in the span is above the median in its row. More formally, let D be a disparity map, and $M(r)$ be the median value in row r in D. Let r_{low} and r_{high} bound an interval of rows, such that in column c, $D(r, c)$ is some d for all $r_{low} \leq r \leq r_h$. The length of the run is clearly $r_h - r_l + 1$. If $r_h - r_l + 1 \geq \ell$, then the span of rows r_{low} to r_{high} in column c is considered an obstacle. The parameter ℓ is a function of $D(r_{low}, c)$ and $M(r)$. A reasonable such function is

$$\ell(disparity, medianDisparity) = \begin{cases} 8 & \text{for } disparity \geq medianDisparity + 2 \\ 20 & \text{for } disparity \geq medianDisparity + 1 \\ 35 & \text{otherwise} \end{cases} \tag{2.5}$$

Algorithm 3 gives a linear-time implementation of this criterion. For each span, a confidence measure is computed. In this implementation, the confidence measure c is determined by the number of pixels in the obstacle n, the number of pixels which are validated m, and the standard deviation of disparity values in the interval s. In particular,

$$c = \frac{m^2}{ns} \tag{2.6}$$

Figure 2.2 shows the output of the algorithm on one image, and Figure 2.3 shows the algorithm's performance on the highlighted column.

The time-complexity of the algorithm is linear. Each pixel need only be examined once. Figure 2.3 shows the computation times of 193 images at 640x480 resolution versus the proportion of pixels in the image which were identified as obstacles.

Once obstacle pixels are detected, bounding boxes are constructed around each connected component of obstacle pixels. A box is classified as an obsta-

Algorithm 3 Obstacle Detection in a Disparity Map

Let $D(r, c)$ be the disparity value of row r, column c in disparity map D
Let $M(r)$ be the median disparity value in row r
Let $O(r, c)$ be 1 if r, c is an obstacle in D, 0 otherwise
for each column c **do**
  Let *State* be `NOT IN OBSTACLE`
  Let *activeDisparity* be -1
  Let $runLength = 0$
  for each row r, starting at the top **do**
    if *State* is `NOT IN OBSTACLE` **then**
      if $D(r, c) = activeDisparity$ **then**
        Increment *runLength*
      else
        Let *runLength* be 0
        Let *activeDisparity* be $D(r, c)$
      end if
      Let $minLength = \ell(D(r, c), M(r))$
      if $runLength > minLength$ **then**
        Let *State* be `IN OBSTACLE`
        Let $O(r - runLength, c)$ through $O(r, c)$ be marked 1
      end if
    end if
    if *State* is `IN OBSTACLE` **then**
      if $D(r, c) \leq M(r)$ **then**
        Let *State* be `NOT IN OBSTACLE`
      else
        Let $O(r, c) = 1$
      end if
    end if
  end for
end for

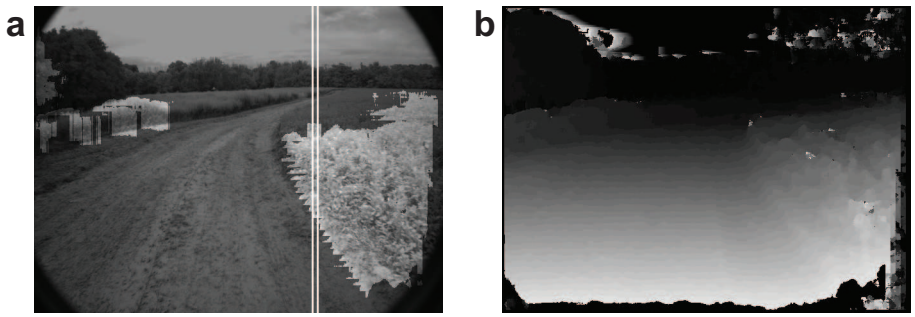

Figure 2.2: (a) Sample scene image, with detected obstacle pixels highlighted and sample column outlined, as analyzed in Figure 4 (b) The corresponding disparity map.

cle if its mean confidence measure exceeds a threshold. A reasonable value for this threshold is 15. Table 2.1 gives the performance of this algorithm on several obstacles at various distances. A ✓ indicates successful detection, and an ✕ indicates that the obstacle was not detected.

Object (height) / Distance	4.5	6	7.5	9	10.5	12	13.5	15	16.5
Short cinderblock (19.5 cm)	✓	✓	✓	✕	✕	✕	✕	✕	✕
Upright cinderblock (40 cm)	✓	✓	✓	✓	✓	✕	✕	✕	✕
Shelves (65 cm)	✓	✓	✓	✓	✓	✓	✓	✓	✕
Trash can (69 cm)	✓	✓	✓	✓	✓	✓	✓	✓	✕

Table 2.1: Detection of objects at various ranges (in m)

As can be seen, the range at which short obstacles can be reliably detected is quite limited. Fortunately the primary function of obstacle detection during the Grand Challenge was to detect graded berms on either side of the course. Data recorded during the Grand Challenge indicates that Prospect Eleven was able to do so at ranges of approximately 8 m, which was adequate for navigation during the race. The limited detection distance for small but dangerous obstacles capped the vehicle's maximum speed.

There are several significant limitations to the algorithm presented. For instance, it is sometimes the case that traversable features have high slopes. Figure 2.3 shows one such image. In practice, such steep hills are rarely encountered, so this limitation was not debilitating.

Another significant limitation is that though actual obstacles are reliably

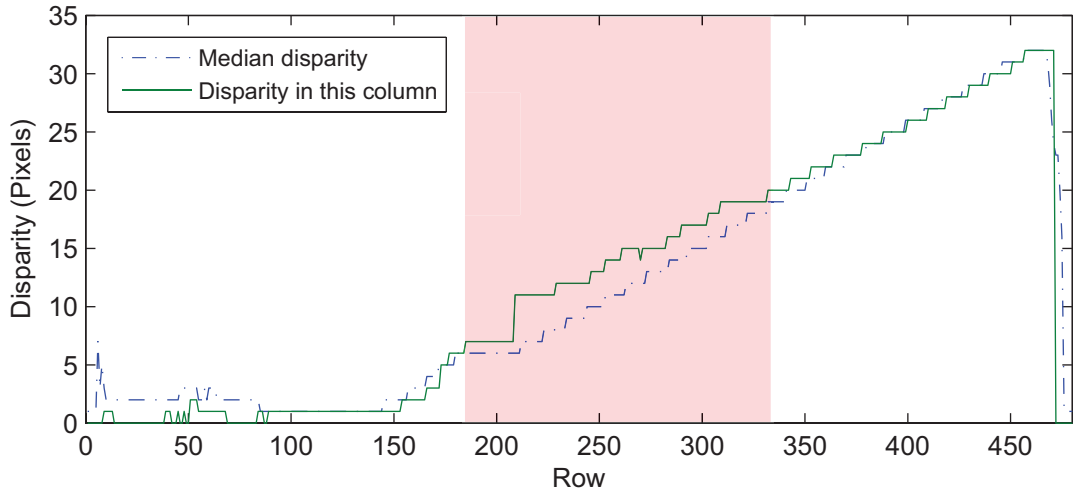

Figure 2.3: Disparity values and obstacle detection in the highlighted column of Figure 3.a. The shaded region is detected as an obstacle.

detected, false positives remains a problem. Figure 2.3 shows such a case. Though careful parameter tuning reduces the frequency of false positive, further reducing the sensitivity of the system could result in not detecting dangerous obstacles. Though the frequency of false positives depends greatly on the terrain encountered, experience suggests that bumpy terrain tends to produce them with greater frequency than does smooth terrain.

A further limitation is the assumption made about a ground plane. When it does not hold that $\frac{\partial z}{\partial x}$ is low, such as a banked curve, the higher part of the curve could be detected as on obstacle. This case was not encountered with frequency during the Grand Challenge.

Perhaps a surprising capability of the system was nigh‘t operation. Beer Bottle Pass was traversed at night, the only light from the headlights of the vehicle. It has even been suggested that night operation is superior for stereo vision, because the close proximity of lights generates very many shadows. These shadows form excellent edges, and improve feature matching.

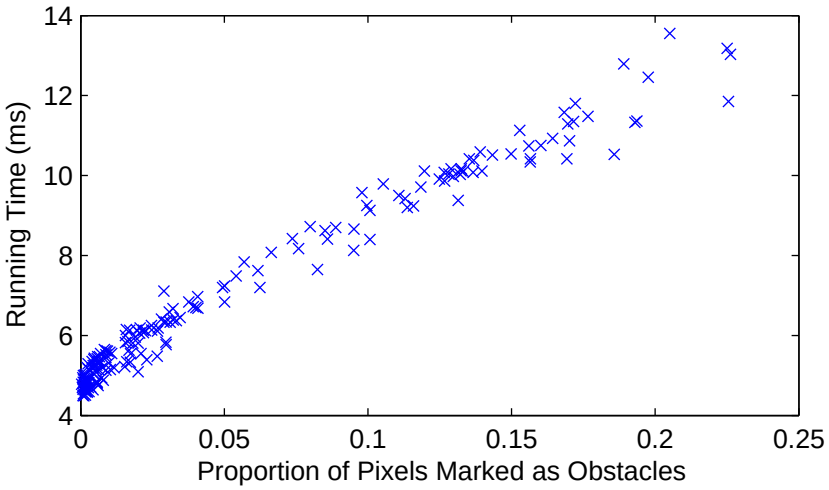

Figure 2.4: Computation Time versus Proportion Obstacle Pixels over 193 640x480 images

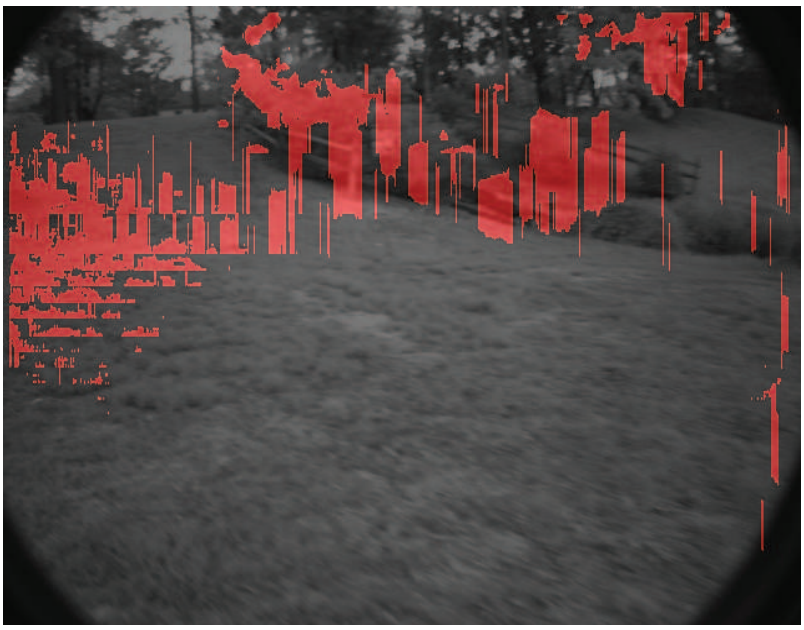

Figure 2.5: False positive: hill

Figure 2.6: False positive: noise in sky

Chapter 3

Filtering in the Time Domain

Detected obstacles are tracked in the time domain to improve accuracy in positioning and limit false positives. When a new image is processed, the list of obstacles from that image is compared to the list of currently tracked obstacles. Each new obstacle is matched to the closest existing one, or declared a new obstacle if no suitable match exists. A confidence measure is maintained for each obstacle, and a Kalman filter maintains the estimate of the obstacle's location.

3.1 Matching obstacles

Matching obstacles is accomplished in a simple manner, and is described in Algorithm 4. Let C be the set of currently tracked obstacles and N the set of obstacles detected in the current frame. Further, let $phi(n, c)$ be the cost of matching some $n \in N$ to some $c \in C$. Then for each obstacle $n \in N$, Algorithm 4 minimizes $\phi(n, c)$ over all $c \in C$. If the $\phi(n, c)$ is below some threshold $maxMatchingCost$, then n and c are matched; otherwise, n is added to a list of new obstacles.

At a minimum, the cost matching function $\phi(n, c)$ should consider the distance between n and c. It makes little sense to match two obstacles who position is very far away. The cost function used by Prospect Eleven is slightly more complex, however. The confidence of c is also considered, as it is preferable to match a new obstacle to an obstacle which is likely to exist. Let d be the distance between c and n, and let $conf_c$ be the confidence of obstacle c. Then the cost function used is

Algorithm 4 Obstacle Matching Algorithm

Let C be the set of currently tracked obstacles
Let N be the set of obstacles detected in the current frame
Let $M(c)$ be the set of obstacles in N matched to $c \in C$
Let D be the set of obstacles first detected in this frame
Let $maxMatchingCost$ be a constant, such that two obstacles can only be matched if their matching cost is less than $maxMatchingCost$
Let $phi(n, c)$ be the cost of matching obstacle $n \in N$ to obstacle $c \in C$
for each $n \in N$ **do**
    Let $minC$ be the minimum-cost match for n.
    for each $c \in C$ **do**
        if $\phi(n, c) < \phi(n, minC)$ **then**
            Let $minC$ be c
        end if
    end for
    if $\phi(n, minC) < c_{maxMatchingCost}$ **then**
        Add n to $M(minC)$
    else
        Add n to D
    end if
end for

$$\phi(n, c) = \frac{d^3}{conf_c} \tag{3.1}$$

At termination of Algorithm 4, $M(c)$ gives a set of obstacles in the current frame which correspond to c, which is currently being tracked. If for some c, $M(c) = \emptyset$, then c was not detected in the current frame. Then the new confidence of c, $conf'_c$, is simply

$$conf'_c = conf_c \cdot atrophyRate \tag{3.2}$$

where $atrophyRate \in [0, 1]$ is a constant. On Prospect Eleven, $atrophyRate = .5$.

Obstacles $c \in C$ for which $M(c) \neq \emptyset$ are updated. If $|M(c)| > 1$, the obstacles are combined by constructing a single bounding obstacle. Now all that remains is to update each the position and confidence of c. Updating the position of c is done by a Kalman filter, discussed in the next section. However, the new confidence is given by

$$conf'_c = \frac{\max(conf_c, conf_n)}{atrophyRate} \tag{3.3}$$

3.2 Maintaining Obstacle Position

The position of each obstacle is maintained using a Kalman Filter. Kalman filters are a well-studied technique for tracking data. For linear systems with uniform random Gaussian noise, it gives the optimal estimate of a quantity given past measurements. This section gives an overview of Kalman filtering following [12], in particular its application to obstacle localization.

In this case, the Kalman filter estimates the value $x \in \Re^3$, where x is the position of an obstacle relative to the car expressed in homogenous coordinates. Homogenous coordinates allow translation to be performed as a linear operation. In particular, homogenous coordinates represent a position x, y with the values x_h, y_h, c. Standard coordinates relate to homogenous coordinates by the equations $x = x_h/c$, and $y = y_h/c$. As is shown later, it is easy to represent translation as a linear operation by homogenous coordinates.

Further, we suppose that the value of x at state k, x_k, is governed by the equation

$$x_k = Ax_{k-1} + w_{k-1} \tag{3.4}$$

The matrix A relates the state at one time step to the state in time step k. In this case, A simply takes into account the movement of the car since the previous time step. The matrix w represents the noise associated with process updates — i.e., the noise associated with the car's own state estimation. It is supposed that w follows a uniform Gaussian distribution centered at the origin with covariance matrix Q.

It is further supposed that measurement of x_k, z_k, is governed by the equation

$$z_k = x_k + v_k \tag{3.5}$$

Similar to w, v is the measurement error. It is supposed that v follows a uniform Gaussian distribution centered at the origin with covariance matrix R.

The Kalman filter maintains the optimal measure not only of x, but also of the covariance of x, P. In particular, given an update in state estimation, a translation matrix A is computed to reflect the rotation and translation of the vehicle. Given a translation vector of (x, y), the appropriate matrix A_t is given by

$$A_t = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x & y & 1 \end{pmatrix} \tag{3.6}$$

The rotational component, caused by changes in the car's heading, is similarly computed. Given a heading change of d degrees, the appropriate transformation A_r is given by

$$A_r = \begin{pmatrix} cos(d) & sin(d) & 0 \\ -sin(d) & cos(d) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.7}$$

The final transform as a result of the car's movement, A, is simply

$$A = A_r A_t \tag{3.8}$$

And thus the update equations are

$$x_k = A x_{k-1} \tag{3.9}$$

$$P_k = P_{k-1} + Q \tag{3.10}$$

where Q is determined based on the error in state estimation.

Computing the update equations for new measurements is much more involved, and full derivations are available in [12]. This paper provides only the final equations, and the aspects which are particular to this implementation.

Obtaining a new estimate of the obstacle's state requires only knowledge of the obstacle's current state x_{k-1}, the covariance matrix representing the error in the obstacle's current state P_{k-1}, the new measurement of the obstacle's state z_k, and finally the covariance matrix R_k representing the error distribution in z. The covariance matrix R_k is computed by supposing that the error in localization is caused entirely be inaccurate and quantized disparity measurements, and that the error distribution for disparity measurement is independent of the disparity value itself. Differentiating Equation 1.2, this yields that the error is inversely proportional to the square of the disparity d.

With this information, a measurement update may be performed. The Kalman gain K is first computed. Then the Kalman gain is used to obtain the new state estimate x_k, and the new error estimate P_k.

$$K_k = P_{k-1}(P_{k-1} + R)^{-1} \tag{3.11}$$

$$x_k = x_{k-1} + K_k(z_k - x_{k-1}) \tag{3.12}$$

$$P_k = (I - K_k)P_{k-1} \tag{3.13}$$

3.3 Results

Matching is effective in improving localization, particularly for obstacles at ranges above 8 m. The measurement error for localization decreases quadratically as a function of range. Though ground-truth data is not available, a direct approach to obstacles at randomized positions is simulated with a wide variety of parameters. Figure 3 gives the mean precision of localization at various ranges over 10,000 simulations. Values simulated include vehicle speeds in the range 6 m/s to 13 m/s, minimum detection ranges between 1.5 m and 5 m, maximum detection ranges between 15 m and 20 m, detection frequencies between 6 Hz and 10 Hz, and disparity calculations with unbiased Gaussian error with σ^2 between 0.05 pixels and 0.5 pixels. The error model

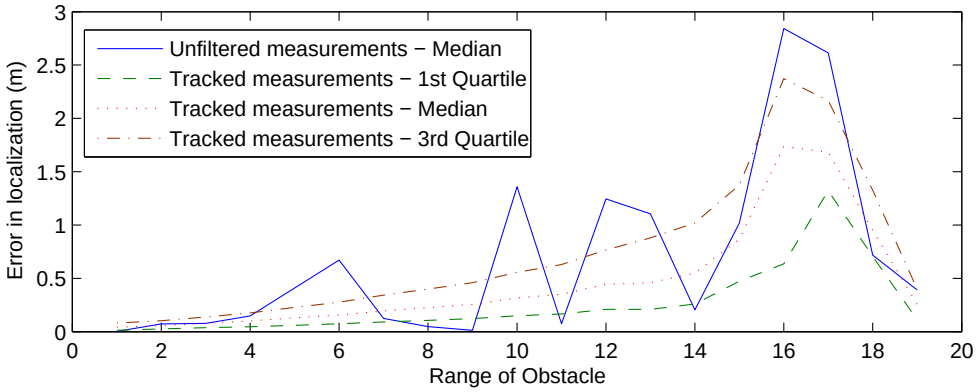

Figure 3.1: Error in localization versus range

assumes that error is caused entirely by miscalculation and quantization of disparity values. The periodic behavior of the measurement error is a result of quantization: for ranges which correspond to nearly integer disparity value, quantization causes very little error. Though there are certainly many more sources of error than those modelled, and actual error is much greater, the simulation demonstrates that tracking is effective at reducing error in localization.

Chapter 4

Monocular Techniques: A Progress Report

One significant disadvantage of stereo vision is its limited range. To augment the range of stereo, this section discusses the possibility of combining monocular techniques with stereo vision. Early results show that the technique has many limitations.

4.1 Color matching

The simplest approach is color matching. Once a obstacle map is obtained, it is easy to determine the distribution of intensity over the pixels marked as obstacles, the pixels not marked as obstacles, and the scene as a whole. Using these quantities and Bayes' Law, it is simple to compute the probability that a pixel is an obstacle, given its intensity. Unfortunately, this technique was not effective. For one, the input data was not ideal, as the stereo system used by Prospect Eleven records only black and white images. Moreover, many obstacles appear very much like objects which are not obstacles — a berm is made of the same dirt and sand that make up the road ahead. Complicating matters even further, shadows in the scene make the color of an obstacle unpredictable. Complete results of color matching are available on the author's website.

4.2 Texture matching

Instead of looking at intensity values, texture metrics consider the way in which intensity values change over some region. In particular, the image is convolved with oriented filters in order to determine the orientation of the surface. A vector is then computed based on the intensity of each orientation in the image. An average vector for both obstacle pixels and non-obstacle pixels is computed. To determine if a new pixel is an obstacle, it is projected onto the vector for obstacle pixels and non-obstacle pixels, and the magnitude of each projection is compared to a threshold. Again, results were extremely disappointing and are available on the author's website.

4.3 Motion estimation

As monocular techniques for image understanding seemed unfruitful, a new area was investigated: motion estimation. During the Grand Challenge, Prospect Eleven relied on a donated Applanix Position and Orientation System. Though GPS is helpful for general localization, it only operates at 1Hz and has significant errors. Another system is needed to estimate small movements between GPS updates.

Fortunately, a great deal of research has been conducted on using inferring motion from a set of images. The advantages to the approach are many; cameras typically operate at high frequency relative to other sensors such as GPS, and they are a consumer good so are very low cost. One of the most promising approaches was developed by [9]. Using Scale Invariant Feature Tracking, or SIFT, features in one image are matched to features in the previous image. This matching gives a transform from one image to the next. Because matches can sometimes be unreliable, a bundle adjustment algorithm is employed in order to reject poor matches. The SIFT matching algorithm appears to be very robust on images taken from Prospect Eleven's vision system, and ongoing work aims to integrate this technology into the state estimation system of Prospect Eleven.

Chapter 5

Conclusion

Perhaps the best lesson of the Grand Challenge was in the importance and difficulty of defining the problem. The system presented in this paper is far from state of the art — its strength and weakness is that it is specifically suited for structured environments similar to those of the Grand Challenge. Instead of focusing on the general problem of autonomous obstacle avoidance, several assumptions were made about the environment ahead, and these assumptions were exploited to design fast, simple and effective algorithms. Moreover, understanding that there is tremendous error in the system motivated an algorithm for tracking obstacles in the time domain, greatly improving accuracy. This system, also simple in nature, dramatically improved results and made feasible what would otherwise have been impossible.

Several tremendous challenges remain. For instance, Prospect Eleven was unable to detect a lake upon its return to the desert. It is not surprising that it was unable to do so — the lake confused even the human drivers in the car, and Prospect Eleven was not designed to detect lakes. However, this one problem symbolizes the enormous complexity of autonomous navigation. In no way could every possible case be considered in code; simplifications must be made to make the problem tractable.

The challenges that remain ahead cover some of the most fundamental problems of engineering: machine learning, adaptive systems, and high dimensional information processing. Though the challenges are enormous, so too is the potential to revolutionize the world of transportation.

Bibliography

[1] A.R. Atreya, B. C. Cattle, B. M. Collins, B. Essenburg, G. H. Franken, A. M. Saxe, S. N. Schiffres, and A. L. Kornhauser. Prospect Eleven: Princeton University's Entry in the 2005 DARPA Grand Challenge. Submitted to the Journal of Field Robotics, 2006.

[2] S. Badal, S. Ravela, B. Draper, and A. Hanson. A practical obstacle detection and avoidance system. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 97–104, 1994.

[3] A. Broggi, C. Caraffi, R.I. Fedriga, and P. Grisleri. Obstacle detection with stereo vision for off-road vehicle navigation. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005. Pages 65–72.

[4] L. Matthies and P. Grandjean. Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. *IEEE Trans. On Robotics and Automation (T-RA)*, 10:783–792, 1994.

[5] L. Matthies, A. Kelly, T. Litwin, and G. Tharp. Obstacle detection for unmanned ground vehicles: A progress report. Proceeding of the International Symposium on Robotics Research, 1996.

[6] Michael Pasqual. *The Vision System of an Autonomous Ground Vehicle and the Policy Implications of Automated Technology for the United States*. 2005. Senior Thesis, Department of Electrical Engineering, Princeton University. Advised by Alain Kornhauser.

[7] Robert Sedgewick and Michael Schidlowsky. *Algorithms in Java, Third Edition*. Addison-Wesley Professional, 2002.

[8] K. Sobottka and H. Bunke. Obstacle detection in range image sequences using radial slope. pages 535–540, Madrid, Spain, March 25–27 1998.

[9] Dennis Strelow and Sanjiv Singh. Optimal motion estimation from visual and inertial measurements. In *Workshop on the Applications of Computer Vision (WACV)*, pages 314–319, 2002.

[10] A. Talukder, R. Manduchi, L. Matthies, and A. Rankin. Fast and reliable obstacle detection and segmentation for cross country navigation. IEEE Intelligent Vehicles, 2002.

[11] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice Hall, 1998.

[12] G. Welch and G. Bishop. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina at Chapel Hill, 1995.

[13] Z. Zhang, R. Weiss, and A. Hanson. Qualitative obstacle detection. Technical Report UM-CS-1994-020, University of Massachusetts-Amherst, 1994.