Dynamic World Modeling Using Vertical Line Stereo

James L. Crowley, Philippe Bobet
LIFIA (IMAG), Grenoble, France

Karen Sarachik
MIT - AI Lab, Cambridge, USA

Abstract

This paper describes a real time 3-D vision system which uses stereo matching of vertical edge segments. The system is designed to permit a mobile robot to avoid obstacles and to position itself within an indoor environment. The system uses real time edge tracking to lock onto stereo matches. Stereo matching is performed using a global version of dynamic programming for matching stereo segments.

1. Introduction

Indoor man-made environments contain many vertical contours. Such contours correspond to environmental structures which a mobile robot may perceive to position itself and to navigate. This system was inspired by the system of Kriegman, Treindl and Binford [Kriegman et. al. 1989]. The stereo system is organized as a pipeline of relatively simple modules, as illustrated in figure 1.1.

2. Detecting and Linking Vertical Edges

The first module in the system is concerned with detecting vertical edges. A cascade of simple filters is used to first smooth the image and then approximate a first vertical derivative. Filters in this cascade are composed of binomial kernel for smoothing, and a first difference kernel for calculating the derivative.

2.1 The Filter for First Vertical Derivatives

Our first derivative filter is composed from a cascade of k convolutions of a circularly symmetric binomial filter, m convolutions of a vertical low pass filter followed by a convolution with a first difference filter.

$$m_{km}(i, j) = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} *k \quad * \quad \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} *m \quad * \quad \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

The results presented below are based on the empirically obtained values of filter of $k = 4$ and $m = 2$.This filter will give a negative response for transitions from dark to light and a positive response for transitions from light to dark. In order to detect points which belong to vertical edges, each row of the filtered image is scanned for extrema. An extrema, or edge point is any pixel e(i, j) that is a local maximum and has more than twice the absolute value of neighbors 3 pixels away.

2.2 Raster Scan Edge Chaining

The second module in the system is responsible for edge chaining and straight line approximations. Raster scan based chaining algorithms are well suited to real time implementation. Raster scan edge chaining is greatly complicated by the presence of edges near the scan direction [Discours 89]. By restricted edges to directions perpendicular to the scan direction the process becomes quite simple. Edge chains are converted to edge segments by the well known "recursive line splitting" process [Duda-Hart 73]. This algorithm is known to exhibit instabilities when representing curved edges. In experiments with curved objects, these instabilities have not proved a problem for subsequent stages in our system.

This work was sponsored by Société ITMI and Project EUREKA EU 110 : MITHRA.

2.4 The MDL Edge Segment Representation

As edge segments are detected they are transformed to a parametric representation composed of the mid-point, direction and length. We refer to this representation as MDL. This representation is designed to facilitate the matching step in the segment tracking phase, and to permit Kalman filter tracking of the center point as two independent parameters. The MDL parametric representation for segments is described in [Crowley-Stelmaszyk 90] in this conference. A segment is represented by a vector $\mathbf{S} = \{c, d, \theta, h\}$ of parameters:

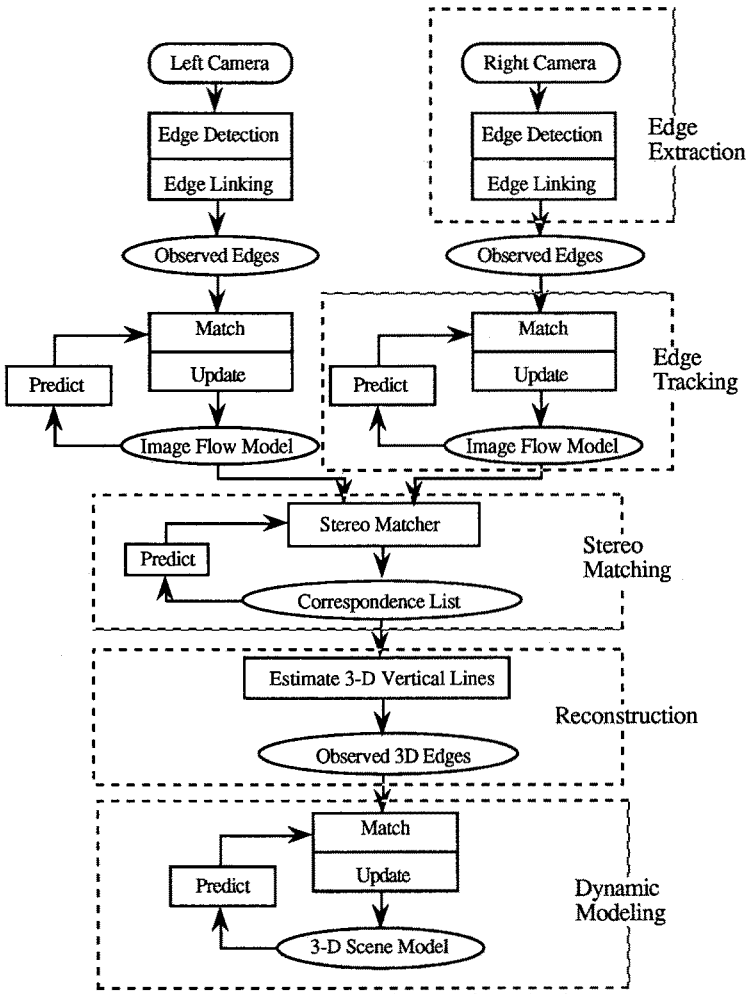

Figure 1.1 The System Organization

3. Measuring Image Flow by Tracking Edge Segments

Tracking allows us to preserve the correspondence between an observed edge and information in the 3-D scene model. This tracking process is well debugged and has been used in a number of projects. Real time hardware has recently been constructed using this algorithm [Chehikian 88]. This process has been described at the second I.C.C.V. [Crowley et. al. 88]. Performing stereo correspondence on the flow model provides cleaner data for stereo matching. In particular, this technique also permits the system to function in the presence of simple occlusions.

Correspondence of edges is maintained by a very simple tracking process based on a Kalman filter. The tracking process maintains a list of "active" edge segments composed of the parameter vector, S ={c, d, θ, h}. The flow model also contains a confidence factor, CF, represented by a state from the set {1, 2, 3, 4, 5} and a unique identity, ID. The identity of a segment permits the process to preserve the association between a segment, its corresponding segment in the other image, and the resulting 3-D segment.

4. Correspondence Matching using Dynamic Programming

Stereo matching is performed by a single pass of a dynamic programming algorithm over the entire image. The process "locks on" to correspondences, by feeding the previously discovered disparity for each segment pair into the matching process.

Line segments from the flow models from the left and right images are combined with predictions from the previous match to produce a new stereo correspondence list. The contents of the stereo correspondence list gives a list of vectors each containing 4 values: (Left ID, Right ID, Disparity, CF). The Left ID and Right ID are the ID's of the matching segments. The Disparity is the most recently observed horizontal disparity in pixels. The CF is the number of times which this correspondence has been found in the last 5 images. IF CF goes to zero, then the correspondence is removed from the list.

4.1 Matching by Dynamic Programming

Stereo matching is performed by a dynamic programming algorithm [Kanade-Ohta 85]. The dynamic programming algorithm calculates the best global match provided that the order of the segments is the same in the two images. The algorithm works by propagating matching costs in a grid. After propagating the cost from the bottom to the top of the grid, the least cost past is traced back to the far corner of the grid. The cells in this path provide the most likely globally consistent set of correspondences of segments from the left and right flow models.

The edge lines in each flow model are ordered from left to right based on column of the midpoint. When more than one segment has the same midpoint, the segments are ordered from top to bottom using the row of the midpoint. Thus, the columns of the DP array grid correspond to the segments from the left flow model, and the rows correspond to the segment from the right flow model. Paths through the DP grid correspond to possible matches of segments. A "lawn-mower" style algorithm is then used to propagate the cumulative cost for every possible path, based on a cost function for each pair of matches.

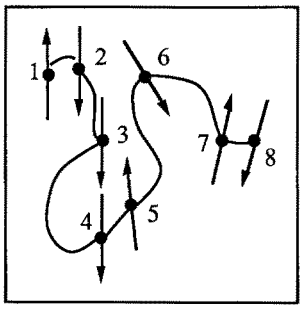

Figure 4.1 The segments in the flow models are sorted by row and column.

4.2 Cost Functions

The individual cost of matching a segment from the left image to a segment in the right image is based on the Mahalanobis distance between the attributes of the segments. Let us refer to S_n as the n^{th} segment from the left flow model and S_m as the m^{th} segment from the right flow model. The individual cost for matching S_n to S_m is given by:

$$C(n, m) = C_o(n, m) + C_\theta(n,m) + C_d(n, m)$$

where:

$$C_o(n, m) = ABS(d_n - d_m) / (h_n + h_m) \qquad C_d(n, m) = (c_n - c_m - D_o) / \sigma_o$$

$$C_\theta(n,m) = (\theta_n - \theta_m) / 10°$$

The term $C_o(n, m)$ is based on the overlap of the segments, as computed from the parameters d and h. If the segments do not overlap, then the normalized difference is greater than 1 and the match is rejected by setting the cost to infinity.

The term $C_\theta(n,m)$ is based on the similarity of orientation of the segments, as computed from the parameter θ. An experiment was performed in which it was observed that the largest difference in angles occurred when a 3-D line segment is tilted away from the cameras by 45°. In this case the observed difference in angle is 10°. Thus the cost for the difference in image plane orientation between the left and right images is given by the difference normalized by 10°.

The term C_d (n, m) is the difference between an expected disparity and the observed disparity. A nominal disparity is initially determined from a fixation distance. This distance is a parameter which can be dynamically controlled during matching. Whenever a stereo match exist from the previous frame has been determined with a CF > 1, the nominal disparity is reset to this previous disparity. In this way the process is biased to prefer existing matches. The cost is determined by dividing the difference from the fixation disparity by an uncertainty, σ_o , which is also a parameter which can be controlled in the process. The cost of skipping a match is equal to the cost from a 1 standard deviation difference on all three measures, that is $C_{skip} = 3$.

The system has been regularly operated using live images during debugging over the last few months. In a typical experiment, a sequence of 5 to 10 pairs of stereo images is a made as the mobile robot moves in a straight line. Matching statistics have been improved from early results of around 80% correct to nearly 100% correct by improving the stability of edge segments that are detected. This additional stability was achieved by increasing the degree of smoothing from m=2 to m=4, as well to enlarging the tolerance for recursive line fitting to 2 pixels.

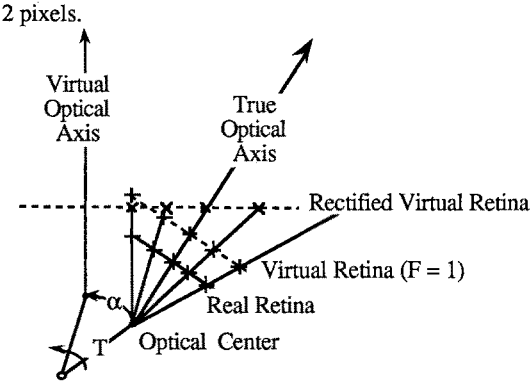

Figure 5.1 Cameras can be mounted with a vergence angle α. Rectification corrects for this angle by projecting back to virtual co-planar retinas.

5 Recovery of 3-D Position from Stereo Information

The 3-D inference process begins by rectification of the position of segments which are found to correspond. Segments are then limited to their overlapping parts, followed by a the calculation of the depth and uncertainty for the end points in a camera centered coordinate system. Segment parameters are then computed from the end-points, and the representation is transformed to scene centered coordinates.

5.1 Projection to a Virtual Retina at F=1

When the retina's of stereo cameras are co-planer, the depth equations reduce to a trivial form. However, it is impossible to mechanically mount cameras such that their retinas are sufficiently close to parallel for this simplification to apply. This problem is avoided by calibrating a transformation which projects points in stereo images to a pair of virtual co-planar retina's as illustrated in figure 5.1. We call these the "virtual rectified retinas. We use a technique inspired by Ayache [Ayache 88] to perform this transformation.

5.2 Depth from Co-planar Stereo Cameras

For coplanar retinas, the depth equation falls directly from a difference in similar triangles. For cameras separated by a base line distance of B, the depth, D, is given by the disparity by $D = F B / \Delta x$. If we observe that our edge lines are rectified to a virtual retina where F = 1, then the depth, D, is given by

$$D = B / \Delta x.$$

The linear term B is determined as a final step in calibration by calculating the distance between the optical centers of the left and right cameras as provided by calibration.

Projection to 3-D is performed using the segment end-points. Because segment length is not always reliable, we must first determine the overlapping part of the corresponding segments. For each of the end points in the corresponding segments, we compute the depth, $D = B / \Delta x$. We then compute the corresponding point in the scene, in the coordinate system of the stereo pair of camera as:

$$x_c = D\, x_{rr}\, (d_i/F) + B/2 \qquad y_c = D\, y_{rr}\, (d_j/F) \qquad z_c = D$$

The uncertainty of each recovered point is modeled as having two independent components: an uncertainty in x, σ_x, and an uncertainty in D, σ_D.

6 Sample Results

This system is the subject of ongoing tests and refinements at our laboratory. A sample of the results form the system is presented in the following figures. This is the second pair from a set of 5 pairs taken at 10 cm displacements. Matches were 100% correct in 4 of the 5 pairs in this sequence, with one incorrect match in the first stereo pair. Figure 6.1 shows the raw images and the vertical edge lines which are detected. Positive edges are shown in white, negative in black. Figure 6.2 shows the correspondance between the segments in the left and right flow models. Figure 6.3 shows an overhead view of the 3D segments which were projected into scene coordinates.

Acknowledgements

The architecture for this system was developed during discussion with Patrick Stelmaszyk and Haon Hien Pham of ITMI, and Alain Chehikian of LTRIF, INPG. The edge detection and chaining procedures are based on code written by Per Kloor of Univ of Linkoping during a post-doctoral visit at LIFIA. The first version of the stereo matching was constructed by Stephane Mely and Michel Kurek. This system is the result of a continuous team effort, and the authors thank all involved for their mutual support and encouragement.

Bibliography

[Ayache 88] Ayache, N. "Construction et Fusion de Représentations Visuelles 3D", Thèse de Doctorat d'Etat, Universtié Paris-Sud, centre d'Orsay, 1988.

[Chehikian et. al. 88] Stelmaszyk, P., C. Discours and , A. Chehikian, "A Fast and Reliable Token Tracker", In IAPR Workshop on Computer Vision, Tokyo, Japan, October, 1988.

[Crowley et. al. 88] Crowley, J. L., P. Stelmaszyk and C. Discours, "Measuring Image Flow by Tracking Edge-Lines", ICCV 88: 2nd International Conference on Computer Vision, Tarpon Springs, Fla., Dec. 1988.

[Crowley-Stelmaszyk 90] Crowley, J. L. and P. Stelmaszyk, "Measurement and Integration of 3-D Structures By Tracking Edge Lines", First E. C. C. V. (these proceedings), Antibes, April, 1990.

[Discours 89] Discours, C., "Analyse du Mouvement par Mise en Correspondance d'Indices Visuels", thèse de doctorat de nouveau régime, INPG, Novembre, 1989.

[Duda-Hart 73] Duda, R. O. and P. E. Hart, "Pattern Classification and Scene Analysis", Wiley, 1973.

[Ohta-Kanade 85] Ohta, Y. and T. Kanade, "Stereo by Intra and Inter Scanline Search using Dynamic Programming, IEEE Trans. on PAMI, 7:139-154, 1985.

[Kriegman et. al. 89] Kriegman, D. J., E. Triendl, and T. O. Binford, "Stereo Vision and Navigation in Buildings for Mobile Robots", IEEE Transactions on Robotics and Automation, Vol 5(6), Dec. 1989.

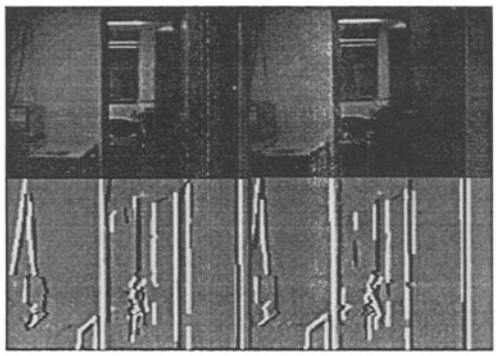

Figure 6.1 An example of the vertical edge segments detected in a laboratory scene.

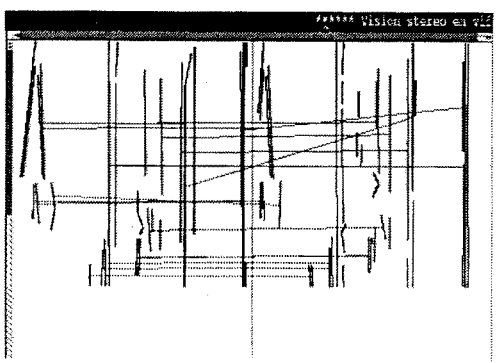

Figure 6.2 Stereo Correspondence for segments extracted from the stereo images.

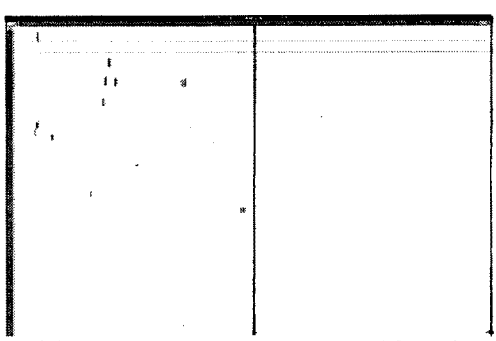

Figure 6.3 Overhead view of the vertical segments reconstructed from the correspondences in figure 6.2.