Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

Non-pixel Robot Stereo

Tyler C. Folsom
tfolsom@ieee.org
University of British Columbia, Vancouver BC V6T-1Z4, Canada

Abstract - The 2005 DARPA Grand Challenge was a 213 Km robot race across the Mojave Desert between Nevada and California. Our team attempted to run the course using only stereo vision. I designed a novel real-time large-image stereo vision system that uses sparse stereo and is biologically inspired. It uses a filter similar to that used in primary visual cortex to locate small line segments. These are then arranged to form a polyline representation of the scene and estimate the depth. The technique was capable of nine frames per second of a megapixel image on a single processor. Accuracy was questionable and the robot did not compete in the race for other reasons.

I. Introduction

Pixels strike mammalian retinal photodetectors, but are immediately transformed and never sent to the brain. Biological vision functions quite well using something like Gabor functions as the primitive. These appear to be the most primitive representation of the input available in cerebral cortex [1] [2]. Yet almost all the research in machine vision uses pixels as the primitive. How could visual tasks be done using Gabor functions as the primitive? I attempted to answer this question with an application to a real world problem: stereoscopic ranging of obstacles in the 2005 DARPA Grand Challenge.

I developed a method of edge and corner detection without using pixels as part of my doctoral dissertation [3]. The method is based on the strategy used in the brain, where pixels strike the retina but are discarded before transmission to the brain via the optic nerve [4]. A truly biologically modeled system would go beyond this signal transformation and continue with the brain's organization, as done in [5] and [6]. Since we are personally quite good at vision we know that this organization works. The interesting question is: does it work on a conventional computer that does not have a billion parallel processors? It has been demonstrated that this method can find edge location, orientation and contrast in artificial images [7]. It has never before been tried on a real world problem. We decided to use this method on the 2005 DARPA Grand Challenge.

A. Biological Vision

This work is inspired by a biological approach but does not attempt to mimic biology. The question that I attempt to answer is "Why does the brain do this transformation and what computations can be based on it?" This can be thought of as fitting into Marr's first two levels: what is the goal of the computation and what algorithms can be used to carry it out [8]? The third level is hardware implementation. I have chosen to implement on a standard digital computer. I will leave it to others to show that the algorithm can be implemented neuromorphically.

B. Edge Detection

The visual field is tiled by a set of overlapping circular receptive fields. Each circle contains four filters with an odd and even pair oriented horizontally and vertically. Each filter is correlated with the image. We make the assumption that at this scale, the image consists of dark and light half planes. By examining the phase of the odd and even filters, we can find the position of the edge. If our assumption that the image is a pure edge is correct, we can find its position to subpixel resolution. The magnitude of the two filters corresponds to the contrast of the edge. Features below a threshold are ignored [9].

The four filter technique is a simplification of the original method. The filters are steerable, which means that the number needed to interpolate the angle is related to the degree of the polynomial [10]. The filters resemble Gabor functions, but they are actually windowed polynomials. The odd filters are first order and the even filters are second order. Five filters are required to find orientations: two odd and three even. Finding orientation takes about half the algorithm's time. For stereo vision, only features parallel or perpendicular to the epipolar line are of interest. Thus for efficiency, we only look at filters in four orientations.

A brief explanation of this method is on the Web at [11]. The phase of the odd and even filter pair is used to locate the edge.

Some definitions:

A *one-dimensional feature* has a significant filter response in one direction but not in the normal direction. This is the same thing as a short line segment.

A *two-dimensional feature* has a normal response that is large relative to the primary response. This may be a corner or other interest point. When orientations are restricted to vertical and horizontal, a diagonal line is a two-dimensional feature.

A *polyline* is a doubly linked list of adjacent features.

C. The Grand Challenge

The 2005 DARPA Grand Challenge [12] was a 213 Km robot race across the Mojave Desert between Nevada and California. All participating vehicles must be completely autonomous and complete the unknown course in less than ten hours. Speeds may reach 100 Km/hr. The course is specified by about 2000 to 3000 rectangular segments given in latitude and longitude coordinates. The race was run on October 8, 2005, with five vehicles completing the entire course. Stanford University claimed the prize with a robot that used five LIDAR systems plus a camera [13].

Most teams entering the Grand Challenge have made use of active signal ranging as a primary method for detecting obstacles. Our Team Sleipnir made the decision to make stereo vision the primary obstacle avoidance method. There are several reasons for this decision:

- We believe that stereo can perform adequately in real-time.
- Our method does not depend on global knowledge and can be done in parallel to hit frame-rate targets.
- Our analysis showed a visual system supporting faster driving speeds than swept ranging.
- Visual systems offer a cheaper solution than some other methods.
- Lack of active signals can achieve stealth and is not as vulnerable to countermeasures.

The system that we developed guided a modified Kawasaki all terrain vehicle (ATV). A high accuracy GPS system was used for navigation [14]. The primary sensor was a Silicon Imaging SI-1280F 1280 x 1024 monochrome camera [15]. We used monochrome because it had higher resolution and color was judged to not contribute significant information. We used a single camera with mirrors set up to provide virtual stereo [16] [17] [18]. This avoided problems with different exposures on the two cameras and increased robustness. The image was split into a top and a bottom image, with the bottom being the true image. The mirrors flipped both images right to left and the bottom image was upside-down. Figure 1 shows a view from Sleipnir's garage. This image has some problems with mirror adjustments. In addition to the intended up-down disparity, there is an artifactual right-left shift in the images, which is not completely uniform. These images were adequate for developing the algorithms.

II. STEREO TECHNIQUES

The methods that we used were based on edge detection and a sparse match of features [19]. Much of the work done in stereo is based on dense image analysis [20]. We expect a sparse algorithm to be more suitable for open-ended conditions in real-time. Sparse algorithms are not compatible with the taxonomy developed for dense stereo. Lacey et al.'s [21] experience on the TINA system has led to the conclusion that by the time of reaching the later stages of scene analysis, all required stereo information is provided by the sparse data. They found that accurate determination of dense 3D data can only be done in the context of knowledge of the scene contents.

There are several stages to the stereo match algorithm. They can be briefly summarized as:

A) Find line and corner segments in the image. This is restricted to a grid that limits how many line segments will be found. Only line segments that have a significant component perpendicular and/or parallel to the epipolar are considered. Typically this reduces the problem from 1.3 million pixels to 2,600 line segments.

B) Determine stereo from isolated line segments. A small window (typically 5 x 5) is applied to the line segment representations of both images and the best correlation is selected. If the center of the match corresponds to a segment in each image, the match is noted and uniqueness enforced.

C) Combine line segments into polylines. Segments with incompatible disparities do not combine. Typically there are about 200 polylines in each image.

D) Match polylines. The individual segments that match are made consistent so that one polyline in one image matches one in another image. This is achieved by merging and splitting the original polylines.

E) Generate a single three dimensional representation of the scene. All depths are adjusted so that the depth of each line is smooth.

F) Format the 3D view in a form that is useful for driving in a smooth path while avoiding obstacles.

A. Line Segments

The orientation perpendicular to the epipolar line contains all the information useful for stereo matching. The other orientation carries no stereo information, but it can be useful for joining stereo segments. If an edge is oriented diagonally, it will produce a response at both orientations and it is possible to determine both coordinates. An edge with components at two

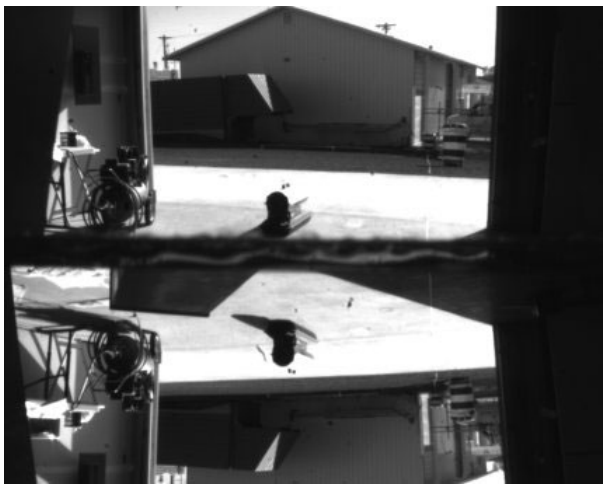

Figure 1. Raw image taken from Sleipnir's garage using single camera with a mirror box.

orientations is called a two-dimensional feature. The original algorithm used steering and determined the primary angle of the feature. In that case, response at the normal angle always meant a corner or interest point. Angle determination has been eliminated because it is computationally expensive. The source code for this method is publicly available [11].

Figure 2 is a detail of the line segments that have been fit to an image. In this detail, a diameter of 21 pixels was used for the filters. The detail is 480 x 300 pixels and is tiled by a 36 x 18 grid. The grid used in the work is hexagonal, but a rectangular grid could also be used.

B. Isolated Stereo

Each position in the grid has been identified as "no feature", a "horizontal edge", a "vertical edge", or "other". To find disparities between the upper and lower

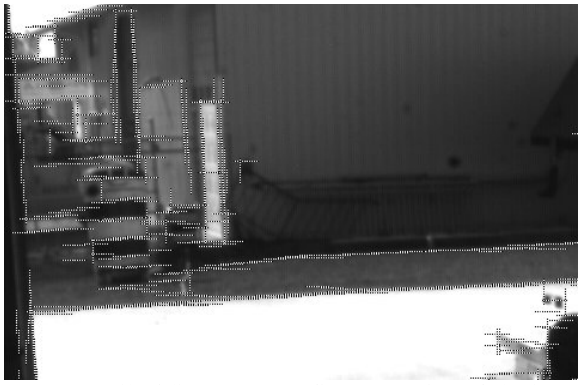

Figure 2. Detail of line segments fit to a portion of the image. The image has been flipped to undo the mirror reversal.

images, form a window with a width of five columns and height three. Compare these windows between images at h positions on the epipolar. Typically we might use h = 5 and circles with a diameter of 21 pixels for a 105 pixel maximum disparity. We find disparity by sliding the window up to h positions and selecting the position that gives the best match to the corresponding feature types in the 5 x 3 slots. If a feature is present, we know which side is dark and which light. Light-to-dark edges do not match dark-to-light edges. Negative disparities are not allowed.

At each of the 15 points in the window, the match is scored according to Table 1. For example, if the lower image has type "horizontal edge" (perpendicular to epipolar line) and the corresponding position in the upper image does also, that position scores +5. That assumes that both features have the same orientation for dark-to-light. If their orientations differ, they are scored as 0. This base score is modified by the difference in edge strength. Contrasts are scaled from -255 to 255. The score found at a point is:

$$\text{Type\_match} - W * |S_u - S_l| / 255 \qquad (1)$$

where S_u and S_l are the contrast strengths in the upper and lower images and W is a weighting factor. A value of 3 has been used for W. Contrast is generally set to 0 for no feature. The score for a stereo match at a point is the sum of these scores over the 15 circle grids of the window.

The point we are starting from has type "horizontal edge" or "other". There are two cases for disparity match. Either the matched point has a significant stereo component in the proper direction or it does not. If it does, the two line segments are mutually marked as matching each other and disparity is more accurately found from the segments. If we do not match another horizontal edge (maybe because of occlusion), we still have a rough idea of the disparity from the window

TABLE 1: TYPE MATCH SCORES; "L" = LINE PERPENDICULAR TO EPIPOLAR; "P" = LINE PARALLEL TO EPIPOLAR; "C" = BOTH; ORIENTATIONS ARE INDICATED BY "+" OR "-".

	none	+L	-L	+P	-P	+C	-C
none	2	1	1	1	1	1	1
+L	1	5	0	0	0	0	0
-L	1	0	5	0	0	0	0
+P	1	0	0	3	0	1	1
-P	1	0	0	0	3	1	1
+C	1	0	0	1	1	5	0
-C	1	0	0	1	1	0	5

match. In either case we can compute the distance as a constant times the reciprocal of disparity.

The stereo match is done twice, once from the lower image to the upper, and once the other way. If both the source and target points are type "horizontal edge" or "other" and they do not match another point better, the results will be the same. In this case, disparity is computed from the positions of the two edges and neighbors do not contribute. In the case of a strong match, the

corresponding matching points are mutually noted and uniqueness is enforced.

We may encounter a situation in which a feature in the lower image matches one in the upper image, but that upper feature matches a different feature in the lower image. This can arise when there is one line in the upper image and two lines in the lower, either due to occlusion or an artifact of edge detection. In this case, pick the stronger match and discard the other.

This initial stereo match is inaccurate. Some points get the correct depth, some are mismatched, and others have a disparity but no definite point of correspondence. Further processing will improve the stereo match.

C. Form Polylines

We start the next stage with a set of three- dimensional line segments in each image. This step is confined to handling each image separately. Polylines are grown by starting with a line segment and considering its neighbors. For example, if we have a horizontal line and are growing to the right, we consider the cells to the right, upper right, and lower right. To add a cell to the polyline it must have a compatible orientation and be close either in disparity or depth. If the disparity of the neighbor is not known, it is acceptable. Cells must be facing in the proper direction to be acceptable. A dark-to-light edge does not join with a light-to-dark edge. Neighbors are selected with a preference to keep the line going at the same orientation.

D. Match Polylines

At this point there is a set of polylines in the upper image and another in the lower image. At features that match, we know the curves of the corresponding segments. For example, lower curve 5 might match five points on upper curve 4, three on upper 84, two on 85, one each on ten other curves and have 34 points with no specific match. It is desired to manipulate things so that each polyline matches only one other polyline. This is done by merging and splitting polylines or by changing the previously identified matches

In the example, we would first see if upper curves 4 and 84 should be combined. Failing that, we would try to readjust the disparities so that the points on lower curve 5 that match to 4 and 84 have the same disparity. If that can't be done, lower curve 5 is split apart into a section matching upper curve 4 and one that matches 84. This process is repeated until curves have a single matching curve.

This process will tend to fragment the curves. These pieces must then be reassembled, mindful of the improved stereo match. A segment that has no definite stereo matches (perhaps because it is parallel to the epipolar line) can be joined with a nearby polyline. Unknown disparities are then filled in from known points on the polyline. Figure 3 gives a depth representation of the lines fit to an image.

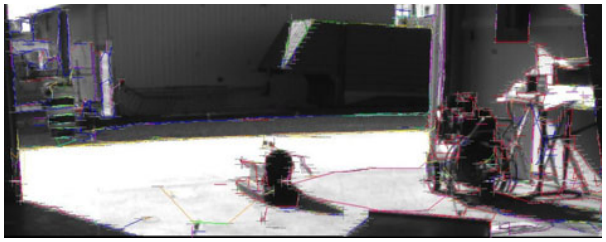

Figure 3. Distances found in an image. It is color-coded so that purples show lines with no disparity, reds show close features, yellow and green are intermediate, and blues are distant. The color codes can be seen in [22].

E. Single 3D Representation

The representation for the top versus bottom image may not be identical to that for the bottom versus top image. If they differ, one must be selected.

It is now possible to convert from pixel locations and disparity to meters from the camera. A simplification step eliminates unneeded points by computing distances. If we are considering three points A, B and C, compute distances AB, BC and AC. If AC is close to AB + BC, eliminate point B [23].

We can generate a polyline that contains vertices in meters. This is compatible with Open GL or DirectX and those rendering engines could be used. It should also be possible to tessellate the surface and give the underlying model. This step has not yet been done. Also awaiting future work are the tasks of formatting the information in a form that supports driving and using multiple frames for motion stereo

III. Timing and Accuracy

The task required for the site visit was to complete a 200 meter course autonomously while avoiding obstacles. The obstacles were to be two trash cans that DARPA people would place at unknown positions on the course. We tested the stereo vision system by taking 25 pictures from Sleipnir's garage. Each picture contained two trash cans at known ranges. The image set contained some difficult cases: cans too far away, too close, in deep shadow or largely occluded. Part of the objective of the photo set was to establish the maximum range for stereopsis. It appears to be about 20 meters. Table 2 gives the known range for each trash can and the median of the range found by the algorithm of objects at the can's location. All ranges are in meters. All images are given in the table. Can numbers may be non-consecutive because some attempts to acquire images failed. The camera was not rigorously calibrated.

The procedure for finding the median was to look at each image and describe manually a rectangular block of pixels containing the trash can. Code was written to

output any range data found within this rectangle and take its median. Features found in this rectangle do not always correspond to the range, which was found with a tape measure. The table shows that there is only a rough agreement between the actual range and that detected by the algorithm. Objects with zero or negative disparity are taken to be at 100 meters. If no object was found in the block of pixels, the situation is reported as "***".

The results in Table 2 were based on using a coarse grid, with diameter of 21 pixels. It is fast, and can do a one megapixel image at 9 frames per second on a 2.4 GHz Pentium 4 PC[1]. We could achieve higher frame rates by using three or four processors.

However, the method suffers from lack of accurate edge location. The base algorithm can achieve sub-pixel accuracy if it is true that it is looking at a tile containing only a half-plane. If the feature is more complex, the algorithm may mislocate by several pixels. The effect is compounded, since large filters are decimated to improve speed. The coarse edge detector is fast, but it is largely limited to determining whether an object is close, mid-ground, or far. Objects that are too close can be ignored, since there is not enough reaction time to do anything. Thus we can use the system as a fast prefilter to identify medium range interest points for further processing.

TABLE 2. COMPARISON OF GROUND TRUTH (TRUE) AND DETECTED RANGES (FOUND) FOR TWO TRASH CANS IN EACH OF 24 IMAGES (DISTANCES IN METERS).

Can#	True1	Found1	True2	Found2
4	9.4	3.9	9.8	3.9
6	3.4	3.4	7.5	3.7
7	3.4	4.3	7.3	3.7
8	6.1	3.3	19.2	29.7
9	27.1	100	12.8	17.5
10	27.1	100	27.4	***
11	20.4	100	27.4	***
12	20.4	100	20.4	100
14	13.4	23.7	16.5	14.2
15	13.4	35.3	16.5	14.7
16	13.4	27.1	16.5	14.9
17	13.7	28.1	15.2	26.2
18	13.7	29.8	13.7	13.9
19	13.7	37.9	13.7	16.9
20	13.7	39.5	3.7	1.6
22	21.9	76.6	23.2	***
23	29.3	***	29.3	***
24	29.3	***	30.5	***
25	29.3	***	45.7	100
26	32.0	100	45.7	100
28	39.6	***	45.7	***
29	11.3	11.4	11.9	***
30	10.1	12.9	11.9	8.3

The half-plane assumption is more likely to be true if the size of the tiles is reduced. Thus it should be possible to use smaller filters and get better results. However smaller filters are more likely to produce artifactual double edges when only one line is present. This can arise if the same horizontal edge is detected in the upper field of one filter and the lower field of another. Edges close to the center of a field are positioned accurately, but those on the periphery tend to be less accurate. Thus recognising when a double detection should be a single or double line is nontrivial.

IV. CONCLUSIONS

The non-pixel method can be used to classify objects as close, intermediate or distant. It is fast enough to be competitive with existing algorithms. It has not yet been optimized and it can go faster. It has the property that its speed can be controlled by making the circular windows bigger or smaller. Thus, while the robot is standing still at the start, it can do a highly detailed scan. When the robot is moving at maximum speed, the level of detail can be reduced to maintain real-time performance. The grid does not have to be uniform; objects representing possible collisions can be handled with more detail than those on the periphery.

The method is highly suitable for parallel implementation and does not depend on global processing. It could be done on three processors, with one handling the center at higher detail and the others covering the left and right periphery.

While the speed is acceptable, the accuracy does not appear to be good enough. Double lines can be produced as artifacts of the method and they are difficult to correct. The line growing and matching is somewhat ad hoc and not as robust as would be desired. While the algorithm can achieve sub-pixel accuracy on artificial images, it seems unable to replicate that in natural images. Testing has been preliminary and a better characterization of the results is unavailable. For these reasons, it has been decided to not use this method for the 2007 DARPA Grand Challenge.

One reviewer commented that the method is suitable for artificial images, but not natural ones because it has no model for noise. It is true that the method can do subpixel edge location in synthetic images and would thus perform better. Natural image noise and clutter is non-gaussian and non-linear. No one has a realistic model for it. It was hoped that fitting filters to sections of a natural image would defeat the noise. This seems to not be the case.

The source code is downloadable [11]. CDs will also be available at the conference.

[1] The image is a pair of 1265 x 400 pixels each.

ACKNOWLEDGMENTS

I would like to thank Leif Anderson and Andrew Bryden of Level 5 CIS, LLC for the time and resources put into Team Sleipnir. I would also like to thank Jim Albers of CoroWare for assistance on the programming.

REFERENCES

[1] J.P. Jones and L.A.Palmer, An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex, *Journal of Neurophysiology*, 58, 1212-1232, 1987.

[2] D.H. Hubel and T.N. Wiesel, Ferrier Lecture: Functional Architecture of the Macaque Monkey Visual Cortex, *Proc. Of Royal Society of London*, Series B, 198 1-59, 1977.

[3] T.C. Folsom and R.B. Pinter, "Primitive Features by Steering, Quadrature, and Scale", *IEEE Trans PAMI*, 20(11):1161-1173, 1998.

[4] E. R. Kandel, J. H. Schwartz and T. M. Jessel, *Principles of Neural Science*, 4th ed, McGraw-Hill, 2000.

[5] M. Mahowald, *An Analogue VLSI System for Stereoscopic Vision*, Kluwer, Boston, 1994.

[6] R. Miikkulainen et al., *Computational Maps in the Visual Cortex*, Springer, 2005.

[7] T.C. Folsom and R.B. Pinter, "Edge Detection from Cortical Filtering", *CESA '96 Vol. 4. Robotics and Cybernetics*, Lille, France, pp. 932-936, 1996.

[8] D. Marr, *Vision*, Freeman, 1982.

[9] T.C. Folsom, "Sparse Scene Sampling for Robot Vision", Proc. of 10th *IASTED Intnl Conf. Robotics and Applications*, 124-129, 2004.

[10] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger, "Shiftable multiscale transforms," *IEEE Transactions on Information Theory*, vol. 38, pp. 587-607, 1992.

[11] http://www.tfolsom.com/Research/Research.htm

[12] www.darpa.mil/GrandChallenge

[13] H. Russell, "Vision Systems Face a Grand Challenge", *Advanced Imaging*, Oct. 2005, pp. 14-16.

[14] Navcom, Technical Reference Manual, 2005.

[15] Silicon Imaging, Inc., SI-1280-CL Manual Rev 1_7, 2004.

[16] S. A. Nene and S. K. Nayar, Stereo Using Mirrors, *Proc. of IEEE International Conference on Computer Vision*, 1998.

[17] S. Baker and S. K. Nayar, A Theory of Single-Viewpoint Catadioptic Image Formation, *International Journal of Computer Vision*, 35(2): 175-196, 1999.

[18] J. Gluckman and S. K. Nayar, Rectified Catadioptic Stereo Sensors, *IEEE Trans PAMI*, 24(2):224-236, 2002.

[19] D.A. Forsyth and J. Ponce, *Computer Vision; A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 2003

[20] D. Scharstein and R. Szeliski, A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, *International Journal of Computer Vision* 47(1/2/3) 7-42, 2002.

[21] A. J. Lacey. et al., The Evolution of the TINA Stereo Vision Sub-System, Tina Memo No. 2001-911, 2002.

[22] http://www.tfolsom.com/Research/Stereo/Fig3.jpg

[23] L.J. Latecki et al., "Construction of Global Maps with Polygonal Objects from Laser Range Data", *Proc. of 10th IASTED Intnl Conf. Robotics and Applications*, 130-135, 2004.