Real-Time Natural Scene Analysis for a Blind Prosthesis

Michael Deering

Computer Science Division, Department of EECS
University of California, Berkeley
Berkeley. California 94720

Carter Collins

Smith-Kettlewell Institute of Visual Sciences
San Francisco. California 94115

ABSTRACT

A real-time computer vision system designed for the limited environment of city sidewalks is presented. This system is part of a prototype mobility aid for the blind. The overall device endeavors to keep blind pedestrians on a safe path down the sidewalk, and also warn of upcoming obstacles. The scene analysis algorithm uses semantic models of the environment to interpret edges in the multi-frame image data as borders of various objects, as well as to assign distance estimates to these objects. The input is a 64 by 64 by 6 bit gray-scale image taken from the vantage point of the shoulder of a pedestrian once a second. Along with each image, the three dimensional transformation of the camera location since the previous frame is assumed to be provided by hardware. After an initial segmentation into edge lines represented as arcs of circles, predictions of edges (generated by analysis of previous frames) are used to identify edges in the current frame. Edges not identified by this process are incorporated into the portion of the three dimensional world model that they are the most consistent with. The induced three dimensional world model of objects can then be used to provide mobility information to the blind user. The emphasis throughout the system has been on efficiency. The design trade-offs and techniques used to obtain high processing rates are discussed. Most of the vision system is currently running in real-time on a 16 bit micro-processor. Field trials of the complete prototype device will begin soon.

I Introduction

An effort to produce an optically based electronic mobility aid for blind pedestrians has led to the development of a natural scene analysis program for the typical scenes encountered by a pedestrian. The restriction to the semanticly rich domain of city sidewalks has allowed the visual processing to be performed in real time on a 16 bit microprocessor. The nature of the task is such that perfect object detection and recognition are not required, but rather the probabilistic detection of potential obstacles. The main goal of the system is to determine approximately where the sidewalk is (with respect to the user), and secondarily to warn of objects blocking the path ahead. This task must be performed in real-time on real-world sidewalks.

Most real-time vision systems to date have dealt with very constrained image domains, due to the enormous computational requirements inherent in visual processing. These include industrial parts recognition [l], blood cell counting, and automatic navigation. Faster hardware and more efficient software techniques will gradually allow more complex domains to be handled at high speeds. Our approach has been to start with very fast segmentation, and amortize semantic processing over several frames, utilizing predictions from models of previously recognized objects to guide the parse of the current image. At the high end, our system is similar to many semantically oriented systems, such as [2] and [3]. Finally our use of multi-frame data is similar to many aspects of [4] [5][6].

II Constraints

In order to achieve real-time processing, we have had to impose several constraints upon the operations of the system:

1. Input is restricted to clean, sun-lit. mostly shadow-free sidewalks.
2. Certain initial starting conditions will be supplied to the system from the outside (which way the camera is pointing, where the sun is, etc)
3. False positives are allowed (it is OK to occasionally warn about non-existent obstacles)
4. We must accept that within our resolution and processing time certain classes of objects are undetectable. These include objects whose width falls below the Nyquist sampling rate of the camera (mainly skinny poles), and objects with very low contrast with respect to the background, or those against a wildly changing background. At the same time, we wished to construct the program modulely.

   allowing knowledge about objects and scenes to be separated from the control structure (but without sacrificing efficiency.)

III Overview of the Program

The input scenes are successive 64 by 64 by 6 bit gray wide angle images taken from the vantage point of the shoulder of a pedestrian, at a rate of one or two frames per second. This relatively low resolution is the highest possible under the hardware and timing constraints. The overall organization of the program is: After video acquisition of the input scene, digitization and noise-removal, the information processed in three passes as follows:

• This work was supported by NSF grant no. PFR-7906299 from the Science and Technology to Aid the Handicapped Program.

1. segment the picture into linked chains of edges,
2. fit curves to these chains and put the mathematical description of the curves into an associative database, and
3. match these curves against several data bases (the world model) which include curve predictions from previous scenes.

The results of these matches identify semantically the objects belonging to the curves. Knowing whether an object is horizontal or vertical allow one to project the curves out into three dimensional space to determine their direction and range. Further heuristics are employed that utilize location information from previous frames to make an independent motion stereo based estimate of the object ranges. At this stage the program should know where the sidewalk and any close obstacles are located, and can proceed to output this information. (Obstacles are the common sorts of large physical objects that one might encounter on a city sidewalk: phone poles, lamp posts, Are hydrants, trash cans, sign posts, automobiles (off to one side), parking meters, trees, bushes, etc.)

IV Coordinate System

The coordinate system used for the three dimensional outside world is centered on the focal point of the camera as it moves through space. The Z-axis is oriented in the direction that the camera is pointing, the Y-axis points straight up, and the X-axis points to the right of the camera. Thus the three dimensional location of all objects is always determined relative to the pedestrian, and are re-computed each frame. Points in the world are mapped to points in the image plane through the usual projective geometrical equations.

One of the fundamental problems of computer vision is that this projection of the three dimensional world (X.Y.Z) to the image plane (x.y) cannot be reversed without additional data of some kind. One of the main goals of the semantic phase of our system is to provide this additional data via semantic knowledge about the probable locations, orientations, and relationships between typical objects encountered within the sidewalk environment. This additional information usually is in the form of a hypothesis on the value of one of X.Y, or Z. Given this value, along with the image plane feature location (x,y), the remaining two three-dimensional coordinates can be found by suitable manipulation of the projection equations.

The motion of the camera in the world between frames will cause the projections of edges of three dimensional objects onto the image plane to change. The general case of the camera transformation involves six parameters. (AX.AY,AZ,t),^,p). For a camera mounted upon the shoulder of a pedestrian it can be safely assumed that AY and *p* are approximately zero, as there is little torsional rotation, and the height of a particular pedestrian remains roughly constant. (It should be noted that the mechanics of the human visual system goes to a great deal of effort to keep *p* near 0, a *p* rotation of up to six degrees of the head is countered by an opposite rotation of the eyeball in the socket. The shape of the horoptor in many animals indicate that the height of the animal is taken to be a constant for some visual processing.) The equations to perform this transformation can be combined with the image plane projection equations to obtain the location within the current image plane of a world point from a previous frame.

V The World Model (the Sidewalk World)

Our world model is the typical sidewalk environment as encountered by a pedestrian. Most modern sidewalks are constructed of slabs of white concrete, and are three to twelve feet wide. Many run in straight lines for an entire block before ending in a corner, while others may be curved. For simplicity it is assumed that all sidewalks encountered by the vision system are straight for thirty feet beyond the camera unless a corner is ahead. Sidewalks mainly differ in their width and the presence (or absence) of a grass border on their street side. These variations are modeled by a few simple parameters.

Within the 64 x 64 image the borders of the sidewalk on either side will often appear as high contrast lines. These lines will be in the lower half of the image, at a highly inclined angle. Various objects bordering on the sidewalk sometimes are of similar optical intensity, reducing the contrast of the sidewalk edges. A large variety of objects can border city and suburban sidewalks. These include: bushes, shrubs, trees, grass, dirt, driveways, walkways, walls of all types, doors, and windows. On the street side usually one finds pavement and automobiles. These objects occur at fairly predictable locations, and many times with good contrast compared to the white sidewalk.

Most objects located upon the sidewalk proper have three fortunate properties: they do not move, have stereotypical locations with respect to the edge of the sidewalk, and usually do not block the path. Objects in this class include: phone poles, lamp posts, fire hydrants, traffic signs, parking meters, trees, mail boxes, phone booths, most trash cans, and bushes. Many of these objects also have the property of being rectilinear, and approximatable as cylinders or boxes (and thus produce good, high contrast edges.)

Unfortunately other objects are more unpredictable. These include: paper boxes, bags, newspapers, trash, garbage cans, parked bicycles, and badly parked cars. Such obstacles can appear anywhere on the sidewalk, and are not always very rectilinear. However, they do have some properties that facilitate their detection. Many are short, so their borders are within the two edges of the sidewalk. They also rest directly upon the ground, enabling their distance to be accurately determined and verified over several frames.

Finally there is the class of moving objects which are the hardest to handle, as their shape and location may change drastically from frame to frame. This class includes: pedestrians, dogs, bicycles, occasionally a car crossing the sidewalk, and wind-blown trash. Fortunately, most mobile obstacles are alive or controlled by humans, and will try not to collide with a pedestrian.

VI Low Level Processing

Initial segmentation of digital images is now a fairly well developed art. but no general technique is know to be (close to) optimal for the large class of natural images, and the speed of various algorithms can differ by orders of magnitude. The necessity for real-time performance is the most severe constraint imposed upon our system. Many promising segmentation techniques had to be rejected out of hand on efficiency grounds.

The speed of the initial segmentation algorithm dominates the performance of the overall system, as the semantic phase is usually many times faster [1]. Thus a poor quality (but fast) segmentation algorithm may be preferable to higher qualify (but slower) algorithm if one can make the semantic phase work a little harder. This is the case in our system. Our segmentation algorithm only directly compares two pixels at a time, and thus is sensitive to noise, but runs at a very high speed. In some sense every module in the system after the pixel comparison has some component who's job is to help correct for the defects introduced by the initial fast segmentation.

The segmentation algorithm used is an edge following algorithm that differs from the usual (such as described in [7]). in that we follow several edges simultaneously. Most edge followers grow an edge line point by point serially from one end of the line. We instead grow many edge lines in parallel by adding to both ends of many edge lines simultaneously. The advantages of this method corresponds roughly to those gained by a breadth first versus a depth first search, in that there is more global information available when one is forced to make local decisions. This allows edge thinning to take place at the same time as edge following, and contributes to the speed of the algorithm.

In more detail, edge points in the input are found using a pseudo-random scan [8]. In the area around each point a search is made for existing edge lines and other isolated edge points. Based on complex decision rules, an existing edge line may be extended to the new point, a new edge line may be created between the new point and another point, or two edge lines may be joined through the new point. (These rules are similar to those found in [9], but with less complex weightings.) The decision rules mentioned above help thin the edges, mainly by forcing edge lines to be essentially continuous. The final result of pass 1 is the collection of edge lines obtained after all the edge points have been processed.

The edge follower tends to be conservative, as it knows that pass 3 will connect broken lines. This is possible as pass 3 has access to more global information about the objects within the scene, and thus may have reason to believe that three roughly collinear line segments may in fact be the edge of one object. Pass 1 does not have enough information to decide if a gap between two line segments is due to noise breaking up a single edge, or is really an occluding object or a gap between two objects.

In our system, noise in a single pixel many times can lead to the break-up of a potential edge line into two pieces. The defects in this edge segmentation can be modeled as higher level noise. That is. as broken edges, missing edges, and misoriented edges. Semantic rules about line segments can take these defects into account, and sometimes even make use of certain properties of the "noise". For example, the breakup of a edge line corresponding to the edge of an object in the scene may be caused by a surface discoloration near the object edge. If this is the case, then the "noise" will be serially correlated from frame to frame. Thus the broken edge will be broken in the same way in several frames, and the relative location of the break can be (and is) used as a feature of that edge (which can help to re-identify it in successive frames.)

VII The Fitting of Edges to Curves

Pass 2 gathers information about each edge line, summarizes its attributes, and sorts it to permit quick searches. Pass 1 sorts the edge lines by x-y location and computes their length. Pass 2 computes their "curvature" and angle of inclination from the horizontal, and sorts them by angle. Edge lines that are too convoluted are broken up into smaller (and simpler) segments. This covers the few cases in which the conservative edge follower described above is not sufficiently conservative. This can occur when two straight line segments intersect at a shallow angle. Pass 1 cannot distinguish this case from a single shallowly curved line segment. Pass 2's curvature statistics are needed to resolve this case.

We devised a fast algorithm to fit lines to arcs circles. The main point for our application was to within a milli-second classify a given line segment as a roughly straight line, a gently curving line, or noise. It is possible that in the future we may make use of finer details of the curves, but in an environment of rotating three dimensional objects absolute curvature is of little use. and more general information on object surface orienta tion and distance is needed (such as discussed in [10].)

VIII Representation of Objects

Our three dimensional representation necessariy emphasis the edges of objects, given the nature of our initial segmentation. The representation roughly resembles a collection of three dimensional edges of the object, but the locations of the edge lines relative to each other is not fixed as in 3D wire frame models in computer graphics, but rather are allowed to vary as needed. As most objects in the sidewalk world are somewhat rectilinear, many are represented by planes parallel to the X.Y or Z axis (a similar representation was used in [4].) For example, a phone pole can be fairly well approximated by a rectangle facing the observer. The sidewalk proper is approximated by a rectangular slab who's position and width is updated every frame with new data.

To achieve high processing speeds, some of our representation is procedural rather than semantic But. as we have built up the number of objects that we nan dle, a number of common subroutines have emerged, allowing new objects to be added and represented fairly easily. High processing speeds verses separation of control structures and knowledge are not necessarily incompatible, but to obtain both one must have intervening software step that transforms high level abstractions into a form combinable with control structures. It also helps to have a very flexible control structure. Our system puts both edge data and object building procedures into associative data-bases, thus allowing the flow of control to be determined by the data. In retrospect, most of the object handling procedures could have been generated by machine from static descriptors rather than by hand, and we may go to such a system in the future.

DC Representation of Visual Knowledge

With the knowledge of how to recognize and represent objects handled by the object representation, the remaining visual knowledge of interest is that that tells you *where* an object is (it's distance and direction.) A number of hueristics of varying degrees of generality exist to do this job. with varying degrees of accuracy and constraints. These are:

1. If the object is know to be resting on (or very near) the ground plane, then Y is known to be -UserHigth, and X and Z can be obtained by back projection.
2. If the object has a known distance from the edge of the sidewalk (for example, phone poles are usually one foot in), and all one has is a piece of an edge of the object (usually not the ground plane intersection), then one can obtain the objects (X.Y.Z) location as follows: take the image plane (x.y) location of the edge piece, project it as a line through the origin (the focal point of the camera) into (X.Y.Z) space, and intersect this line with the plane which is the constant distance in from the sidewalk. This intersection X and Z will be the object's location on the ground plane. (The equation of the plane parallel to the sidewalk is obtainable because the equation of the sidewalk edge is assumed to be known.) Even if the constant distance in from the sidewalk edge is incorrectly guessed, which can lead to distance error on the order of 50% or more, at least some distance information has been provided, and one can make simple decisions like "will 1 run into it in two seconds or twenty seconds". In future frames, motion stereo can tighten up this distance estimate (and correct the constant distance term). By the time an object initially sighted twenty or more feet away comes to within five feet of the pedestrian the location of the object will have appeared in thirty frames of solid data, hopefully pinpointing it's location to within a foot.
3. Once the effects of camera tilt and pan have been subtracted, the differences in locations of a feature in successive frames can be used to determine its range and distance by working backwards from the projection and camera transformation equations. "We employ motion stereo as a secondary distance cue that is used to check up on our primary cues 1 and 2 above.
4. There are some distance heuristics that are only of use for determining the equation of the sidewalk's borders. These include making use of the known constant width of the sidewalk.
5. Finially, the location of an image feature relative to the current, interpretation of scene can be used to obtain a probable distance estimate. For example, edges near the vanishing point of the sidewalk are probably (but not necessarily) far away. Edges way off to one side and in the sky most likely belong to upper stories of buildings or the background, and may be safely ignored. (One misses overhangs this way, but overhangs in general are very hard to recognize,.as many are of very low contrast to begin with.)

X Semantic Analysis

The semantic phase endeavors to build a three dimensional model of the outside world that it is moving through, such that edges in the image frames correspond to edges of objects in three dimensional model. The various distance hueristics listed above are employed both to initially place objects as well as to verify their location/identity over several frames. (An object whose distance varies wildly from frame to frame may be mis-identified.) Further constraints exist that simplify the semantic analysis task. These are:

1. Most objects in the sidewalk world rest on the ground plane (though we do not assume that their point of contact is visible.)
2. Most objects can be roughly approximated by planes parallel to the x, y. or z axis (as in [4].)
3. Location accuracy need only be enough to avoid objects *most* of the time. For example, distances to objects need only be computed with an accuracy of ±20% when objects are closer than 8 feet, and ±40% when objects are further away.
4. The camera transformation will be correctly supplied most of the time (by hardware) to within 1% angular accuracy and 10% translation accuracy.

In order to speed the identification of edges in a new frame, predictions of edge locations from previous frames are used. Much of the speed of the semantic pass is due to the essentially hardware solution of the successive frame registration problem. Most re-occurring edges can have their location in successive frames determineed to with a few pixels be using the camera transformation. The overall effect is a sort of "bootstrapping" re-identification of scene features, similar to that described in [II]. (It may be possible that in the future we can dispense with the special camera motion tracking hardware and recover this information incrementally from optical flow.)

In more detail, the semantic phase is broken up into several subparts. These are:

1. Edge lines from the previous frame are first transformed by the camera transformation and then matched against edge lines in the current frame, current lines that appear to be direct descendents of previous lines are removed from the current data base as "explained". The order in which old object's edges are searched for is determined by their relative semantic importance and data quality. Thus the edges of the sidewalk are usually searched for first, followed by the other objects roughly ranked by their number of (visible) edges.
2. The matches made in 1 induce new information that can be used to construct an updated three dimensional model of the objects that the edges belong to. These models can then make claims for gaps in their edge outlines.
3. The claims made in 2, along with various generic claims for new objects are matched against the remaining data base of edge lines. Residual lines will be claimed as background noise.
4. New object edges obtained in 3 allow for further updating of the three dimensional models, which at this point can be used by the blind navigation system. Predictions and search scheduling for the next frame are made at this point. Objects who's existence is no longer supported by the edge line evidence are deleted in favor of more consistent interpretations.

Thus at any one point in time the world model data base of the system contains models of several objects (phone poles, bushes, automobiles, etc.) that are moving by. as well as a model of the sidewalk proper.

XI Experimental Results

Figure 1 is a sample image taken from an Bmm movie of a sidewalk. One of our test sequences consists of 30 digitized images from this movie. The forward motion between each frame was one and a half feet. On our half speed XL68.000, passes 1 and 2 can process this movie at the rate of one second per frame. The semantic processing of pass three takes an additional half second per frame. When applied to this movie, the system correctly discovered and tracked the sidewalk edges, as well as edges of several objects off to the right of the sidewalk. No objects were found to be blocking the sidewalk. Figure 2 displays a digitized image from the middle of this sequence with the wire-frame model of pass 3 superimposed. (A similar fit is made for each frame of the movie.) A computer animated reconstruction of the outside environment given the world model produced by pass 3 is seen in figure 3. (The detail on the parked car is simulated.) We expect to be running field trials of the entire system in a portable cart trailing behind a blind subject shortly.

XII Incorporation into a Blind Aid

The overall functioning of this system as a blind aid is part of the lineage of a large number of previous tactile blind aids devices designed over the last twenty years [12][13][I4). The computer vision component and the blind interface component have been separated out from each other via the following reasoning:

1. Assuming a perfect computer vision system that knows where every object of interest is located, how could one best communicate this information to a blind pedestrian? What sort of user interfaces and interactions will allow the user to make rapid, accurate use of the information provided?
2. How does one build a portable (wearable!) computer vision system that will locate at least the majority of the objects of interest?

Our solution to 1 has been presented in the previous portions of this paper. Our solution to 2 is to use two output channels - stereo synthetic speech for cognitive (high level) information, and a linear array of 18 tactile elements as a pointing device. The stereo speech unit is a combination of a normal speech synthesis system with a audio processing unit that can "throw" the computer's speech, allowing it to appear to come from a particular direction and distance. (For example, a phone pole might seem to announce "phone pole".) The tactile array is a skin tapping device worn as a head-band, each element corresponds to a particular angular direction, and the frequency of taps of an element corresponds inversely to the distance of the feature being pointed to.

Currently we intention to have the speech unit make major announcements (the blind don't want it babbling all the time, they listen to sound shadows and street sounds.) The tactile output will be used for communicating more mundane information, such as "you're veering off to the left of the sidewalk, veer a bit to the right", by "flashing" the edge of the sidewalk on the appropriate side of the tactile display, should the user veer toward it In any case, one of the main reasons for putting the whole system on a micro processor, rather than simulating it on a mainframe, was to have the capability of expermenting in the real world with various blind interface systems. Also, despite years of experience in testing blind aids, it is very hard to tell how the blind will react to a particular device without letting them make extensive use of it under real world conditions

XIII Perspectives on Future Directions

Within the hardware and timing constraints imposed, we feel that the current system performs well. and cannot be much improved upon. However, for use in a robust blind aid. the system has several limitations which must be overcome. These include the low sensitivity to low contrast edges in shadowed or complex scenes, and the low resolution (""1 degree/pixel) More important is the limitation to processing of edge data only, at the exclusion of surface data. It would be nice to have more information about the sidewalk than just where it's edges are. such as how flat it is, are there any broken sidewalk slabs or holes[9] Also, the current system must treat any high contrast edge on the sidewalk as a potential object edge, even though most are only flat shadows or stains.

Various surface processing techniques proposed in the literature can solve many of these limitations Texture gradients [15] should provide a fairly robust broad classification of the scene into flat and upright surfaces Optical flow can provide approximate distance estimates. Luminance gradients could indicate surface curvature, which could be used in identifying (and segmenting) phone poles, walls, automobiles, etc. [10). Finally, stereo gradients can not only determine general distance estimates, but for the special case of the almost flat sidewalk, they can be tunned to spot vertical devia-

Figure 1. Origional image.

Figure 2. Digitized image witb wire-frame model.

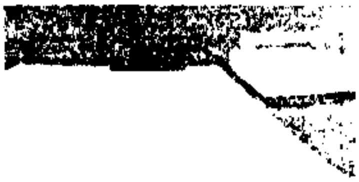

Figure 3. Computer animated reconstruction from model.

tions as small as half an inch (such as un-even sidewalk tiles that one might trip upon). More importantly, stereo can determine that a dark patch *is* flat, and can be safely ignored. This would be similar to the system described in [16]. However, for this specialized stereo algorithm to work, it must have a very good estimate as to where the flat sidewalk is in the first place. This is where the other surface processing techniques enter the loop. Such a system could provide very robust performance under even extreme conditions (such as wet (and reflet live!) sidewalks in the rain), but at the expense of special purpose hardware.

Currently we are in the initial stages of designing a surface processing oriented version of our system system along the lines described above, which is to be implemented in VLSI. This system will be characterize by higher resolution (with separate foveal and peripheral resolution), higher frame rates (approaching 30 frames a second), stereo processing, and extensive use of surface processing techniques. It will differ from other vision systems in that it will be optimized for the sidewalk environment. For example, the stereo section will not have to deal with the stereo frame registration problem in its general form, but for the much simpher case of extracting the (mostly flat) ground plane. Evidence indicates that the vision system of many animals (including man's) has a built in special case solution for extracting the ground plane, which is similar to our proposed technique.

REFERENCES

[I] Perkins. W. A. "A model based vision system for industrial parts." *IEEE Trans Comput* vol. C-27 (1978) 126-143.

[2] Brooks. R., Gremer. R. and Binford. T. "The ACRONYM Model-Based Vision System." In *Proc IJCAI-79* Tokyo. August. 1979. 105-113.

[3] Saphiro. L.. Moriarty. J.. Mulgaonkar. P. and Haralick. R. "Sticks. Plates, and Blobs: A Three-Dimensional Object Representation for Scene Analysis." In *Proc First NCAI* Stanford.CA. August, 1980. 26-30.

[4] Williams. T. "Depth from Camera Motion in a Real World Scene." *IEEE Trans Pattern Analysis and Machine Intelligence* vol. PAMI-2, no. 6 (1980) 511-516.

[5] Tsuji, S.. Osada. M and Yachida. M. "Tracking and Segmentation of Moving Objects in Dynamic Line Images." *IEEE Trans Pattern Analysis and Machine Intelligence* vol. PAM1-2. no. 6 (1980) 516-522.

[6] Roach. J. and Aggarwal, J. "Determining the Three-Dimensional Motion and Model of Objects from a Sequence of Images". University of Texas at Austin Laboratory for Image and Signal Analysis research report TR-B0-2.

[7] McKee, J. and Aggarwal, J. "Finding the edges of the surfaces of three-dimensional curved objects by computer." *Pattern Recognition* vol. 7 (1975) 25-52.

[B] Kavasznay. L., and Joseph. H. "Image processing." *Proc IRE* no. 43 (1955) 56-57.

[9] Prager. J. "Extracting and Labeling Boundary Segments in Natural Scenes." *IEEE Trans Pattern Analysis and Machine Intelligence* vol. PAM1-2. no. 1 (1980) 16-27.

[10] Tenenbaum. J. and Barrow, H. "Recovering Intrinsic Scene Characteristics from Images." in *Computer Vision Systems,* Hanson, A. and Riseman, E. eds., 3-26 (Academic Press, New York. NY, 1978).

[11] Hannah, M. "Bootstrap Stereo." *Proc. First NCAI.* Stanford.CA. August. 1980. 38-40.

[12] Collins, C. "Tactile television: Mechanical and Electrical Image Projection." *IEEE Trans on Man-Machine System* no. 11 (1970) 65-71.

[13] Collins, C. and Madey. J. "Tactile sensory replacement." *Proc. San Diego Biomedical Symposium* Vol. 13(1974) 15-28.

[14] Collins. C, Scadden. L. and Alden. A. "Mobility studies with a tactile imaging device." *Proc Conf. on Systems and Devices for the Disabled.* Seattle, (1977) 170-174.

[15] Witkin. A. "A Statistical Technique for Recovering Surface Orientation from Texture in Natural Imagery." In *Proc First NCAI.* Stanford.CA. August. 1980. 1-3.

[16] Gennery. D. "A Stereo Vision System for an Autonomous Vehicle." In *Proc IJCAI-77,* Cambridge. MA. August, 1977, 576-582.