

Simple Object Recognition and Cognitive Map Formation Using Human-Like Vision in a Virtual World

Ilkay Ulusoy, Ugur Halici
Computer Vision and Intelligent Systems Research Lab.
Middle East Technical University, Ankara, Turkey
Ph: +90 312 2104558
Fax: +90 312 2101261
ilkay@metu.edu.tr
<http://vision1.eee.metu.edu.tr/~halici/>

Abstract. *In this paper we describe an algorithm for object recognition and cognitive map formation using stereo image data in a 3D virtual world where 3D objects and a robot with stereo imaging system are simulated. Stereo imaging system is simulated so that the actual human visual system properties are parameterized. Only the stereo images obtained from this world are supplied to the virtual robot (agent). By applying our disparity algorithm on stereo image pairs, depth map for the current view is obtained. Using the depth information for the current view, a cognitive map of the environment is updated gradually while the agent is exploring the environment. The agent explores its environment in an intelligent way using the current view and environmental map information obtained up to date. Also, during exploration if a new object is observed, from its view from different directions, it is labeled with its shape such as sphere, cylinder, cone ...*

1. Introduction and Motivation

Real robotic applications are very complicated because besides the problems of finding how the robot should behave to complete the task at hand, the problems faced while controlling the robot's internal parameters bring high computational load. Thus, first working in a simulated environment in order to find the strategy to be followed by the robot and then applying this on real robotic applications is preferable. Especially intelligent way finding, path planning and mapping algorithms are developed in simulated environments [1- 5]. In [3] biologically inspired active vision system is also implemented in such a virtual environment.

For complex navigation tasks extending beyond the current sensory horizon, some form of spatial representation is necessary. Higher vertebrates appear to construct representations –sometimes referred to as cognitive maps- which encode spatial relations between relevant locations in their environment [6]. In robotic mapping research, the field of mapping is divided into metric and topological approaches. In topological paradigm [7], environments are represented by graphs. Nodes in such graphs correspond to distinct situations, places, or landmarks (e.g. doorways), which are connected by arcs if there exists a direct path between them. In metric paradigm, grid structure is usually used and environments are represented by evenly spaced grids [8]. Each grid cell may include the presence of an obstacle in the corresponding region of the environment. The best of these two paradigms are combined in [9] where a discrete, 2D occupancy grid map is converted to a topological map.

In very few mobile robotic applications stereovision based mapping and navigation is used because dealing with stereo images is very hard and very time consuming. Despite all the problems, stereovision still becomes one of the most important resources of knowing the world for a mobile robot. The main reason is that imaging provides much more information of the world than most other sensors, although it is very difficult to handle. In most application scenarios, stereovision helps recover depth information of the environment, which can then be used by mobile robots to avoid obstacles, localize itself and recognize visual commands. For example Spinoza uses a trinocular stereo system for such a purpose [10].

In this study we develop an intelligent algorithm to construct a world centric, grid based cognitive map, using human-like stereovision. We develop our algorithm in a simulated world on an agent which has a simulated imaging system with the properties of human eye. Only the stereo image pair obtained from the environment is supplied to the agent. The main achievement of this study

is to use depth information in an intelligent way to construct a grid based mapping in a very short time. The cognitive map obtained here is not a 2D map but a 3D map. Also, while constructing the map, the objects in the scene are viewed from different directions and 3D information about them is obtained. From this 3D information, objects are classified and recognized. In this study the objects in the environment are made up of 3 basic shapes: sphere, cylinder and cone.

The structure of this paper is as follows: In Section 2, the disparity algorithm developed by us is summarized. In Section 3, our simulated world and its properties are explained. In Section 4, calculation of a depth map for the current view of the virtual world and construction of an environmental map from the successive depth maps are given. In Section 5 object recognition is detailed. In the last section results are given and discussed.

2. Depth Perception

The observation underpinning in our disparity algorithm is that there is considerable scope for combining multi-scale phase information to improve the estimation of disparity. Our approach is as follows: We calculate feature points which are located at high contrast edges of the objects in the images by using steerable filters [11]. In finding correspondences, we use orientation and magnitude besides multi-scale phase information calculated at these points in order to avoid the singular points encountered in the method of Jepson [12].

2.1 Extraction of Feature Points

Feature points are found by using steerable filters, as been done in [11], which are attractive as it allows one to work in continuum of orientations. Steerable filters are template filters whose arbitrarily rotated versions can be synthesized by taking linear combinations of finite number of basis filters. In this study, the analytic filter $h(\mathbf{x}) = g(\mathbf{x}) + jq(\mathbf{x})$ (Figure 1) is used as the template where $\mathbf{x} = (x, y)$ is the pixel location in an image $\mathbf{I}(x, y)$. For the multi-scale framework used in this study, $g(\mathbf{x})$ is chosen to be 4th derivative of a Gaussian, which is a steerable filter and $q(\mathbf{x})$ is chosen to be a steerable approximation to the Hilbert Transform of $g(\mathbf{x})$. Here, scaling is done at 3 levels where Gaussian widths are 6, 12, and 24 pixels. For an arbitrary rotation θ , $g_\theta(x)$ can be synthesized using 5 basis filters of $0^\circ, 36^\circ, 72^\circ, 108^\circ, 144^\circ$ as in (1), and $q_\theta(x)$ can be synthesized using 6 basis filters of $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ$ as in (2) where \mathbf{R}_{q_i} is the rotation matrix.

$$g(\mathbf{R}_{q_i} \mathbf{x}) = \sum_{i=1}^5 \frac{1}{5} (1 + 2\cos(2(\mathbf{q} - \mathbf{q}_i)) + 2\cos(4(\mathbf{q} - \mathbf{q}_i))) g(\mathbf{R}_{q_i} \mathbf{x}). \quad (1)$$

$$q(\mathbf{R}_{q_i} \mathbf{x}) = \sum_{i=1}^6 \frac{1}{6} (2\cos(\mathbf{q} - \mathbf{q}_i) + 2\cos(3(\mathbf{q} - \mathbf{q}_i)) + 2\cos(5(\mathbf{q} - \mathbf{q}_i))) q(\mathbf{R}_{q_i} \mathbf{x}) \quad (2)$$

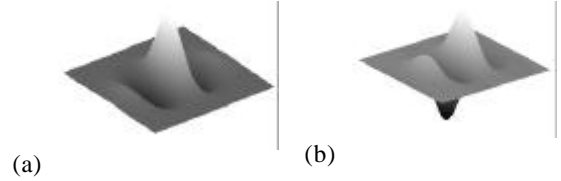


Figure 1 The template filter $h(x)$ used in analysis. a) The real part which is the 4th derivative of Gaussian, b) The Imaginary part which is a steerable approximation to the Hilbert Transform of the real part.

In order to select the feature points, first the image is filtered with basis filters (i.e. $g(x)$ at 5 different orientations and $q(x)$ at 6 different orientations) at 3 different scales and the results are used to interpolate the filtered images of any orientation, θ , between 0° to 180° with 10 degrees of interval by (1, 2). Then, orientation is estimated for each location (3) where $r_{n,q}(x)$ is the response of steerable filtering at scale n and orientation θ and S is the total number of scales, i.e. $S=3$.

$$\tilde{q}(x) = \arg \max_q \sum_{n=1}^S r_{n,q}(x). \quad (3)$$

Then $\left| \sum_{n=1}^S r_{n,\tilde{q}}(x) \right|$'s are thresholded and pixels whose responses are above the threshold are selected to be feature points. Finally, phase (4) and magnitude (5) information at scale n for a feature point at location x is calculated as

$$f_n(x) = \arctan \left[\frac{\text{real}(r_{n,\tilde{q}}(x))}{\text{imaginary}(r_{n,\tilde{q}}(x))} \right]. \quad (4)$$

$$m_n(x) = \left| r_{n,\tilde{q}}(x) \right|. \quad (5)$$

2.2 Finding Disparity and Depth

The attributes used for the correspondence matching of feature points are orientation, magnitude and multi-

scale phase which are calculated as explained in the previous subsection. Since phase is stable with respect to geometric deformations and contrast variations between the left and right stereo views, phase based methods for disparity estimation are successful. However in the neighborhood of singularities as described in [12] phase is not reliable. Thus, in this study, disparity is estimated from the orientation and magnitude similarity besides phase similarity. First, for each feature point at the right (left) image, we search over a window for feature points of similar orientation in the left (right) image. Then the most similar phase-vector among similar oriented candidates is selected to be the corresponding point (6) where magnitude similarity is also checked by using the weighting described in [13]. Here i is for right (left) image and j is for left (right) image. The phase estimates obtained at three different scales for feature point k is given as the vector in (7). C (8) is the weighting used for magnitude similarity check where each element c_{ij}^n (9) shows if the pair magnitudes at scale n are compatible.

$$j = \arg \min_j \left\{ \Phi_i^T C^{-1} \Phi_j \right\} \quad (6)$$

$$\Phi_k = (\mathbf{f}_1(x_k), \mathbf{f}_2(x_k), \mathbf{f}_3(x_k))^T \quad (7)$$

$$C_{ij} = \begin{bmatrix} c_{ij}^1 & 0 & 0 \\ 0 & c_{ij}^2 & 0 \\ 0 & 0 & c_{ij}^3 \end{bmatrix} \quad (8)$$

$$c_{ij}^n = \min \left(\frac{|r_{n,\tilde{\mathbf{q}}}(x_i)|}{|r_{n,\tilde{\mathbf{q}}}(x_j)|}, \frac{|r_{n,\tilde{\mathbf{q}}}(x_j)|}{|r_{n,\tilde{\mathbf{q}}}(x_i)|} \right) \quad (9)$$

This right to left matching algorithm is cross checked with left to right correspondences. In this way we also discard occluded feature points.

The disparity is defined as the pixel distance between corresponding feature points: $\mathbf{d} = x_i - x_j$. In performing this, position shift between the receptive fields of binocular disparity selective cells is mimicked [14]. In this study only the disparity along the epipolar orientation, which is the horizontal in this case, is considered. In order to reach sub-pixel accuracy, a phase shift model of binocular cell receptive fields is inspired from [14]. Here, the sub-pixel disparity is calculated from the inter-ocular phase differences between corresponding points: $\Delta \mathbf{d} = \mathbf{f}_{ij} * \mathbf{1} / 2\mathbf{p}$. Here, $\Delta \mathbf{d}$ is the fine tuning in disparity, $\mathbf{f}_{ij} = \mathbf{f}_i - \mathbf{f}_j$ is the measured phase difference

between the feature points i and j . In this way, the rough disparity estimate found by using position shift model is tuned by the phase shift model.

Depth is calculated using the basic ratio $d=bf/d$ where d is the depth, i.e. the distance of the point from the cameras, b is the baseline (distance between the two cameras), f is the focal length of the cameras and d is the disparity.

3. Virtual Environment

The simulation software of the virtual environment is developed using C++ programming language and OpenGL graphics library on the Microsoft Windows operating system [15]. As the user interface of the software is depicted in Figure 2, it is composed of four panes. In the virtual world shown in Figure 2, there are trees and a cottage among them. The left and right panes above are views from the left camera (eye) and right camera (eye) respectively. These stereo images are used in forming the map and recognizing the objects. The bottom panes render the scene from the top viewpoint, and front viewpoint. These top and front view information are not used in any algorithm but placed here just for visualization purposes. Below of the bottom panes, there are tab based (so called) dialog boxes which allow entering/adjusting of the parameters used throughout implementations.

The software abilities are listed as follows:

1. It allows specifying and rendering 3D arbitrary geometric shapes. The shapes are defined in a text file in a pre-defined format and loaded by the software.
2. The software supports a plug-in mechanism to be loaded and linked dynamically into the application at run-time. This plug-in accesses internal structure of the scene data and images rendered, so that it controls the navigation in the scene by locating the camera and setting camera parameters and exports camera views.
3. We can locate the camera at any point in 3D. This can be fulfilled in three ways. First, at the bottom panes (top view and front view), the user can change the position of the camera and its target by dragging the camera and target symbols within the views. Second, the position coordinates can be typed in the dialog box named "Camera Settings". Third, camera and target locations can be set automatically through the plug-in.

4. Gaze direction, focal length, field of view, base distance between the cameras are the main camera parameters which could be modified whenever desired. This is done either by entering the camera parameters numerically in dialog box named "Camera Settings" at the bottom tabbed control or by automatically setting in the plug-in.

5. Images rendered at the snapshots viewed by the camera(s) (left and right) are exported to be processed for depth perception.

The plug-in used in this study controls the camera position, orientation and parameters. It also receives stereo images rendered at the snapshots. After processing of the images according to the purposes, the plug-in updates the camera location and parameters for the next image rendering in an intelligent way for the purpose of cognitive map construction and object recognition.

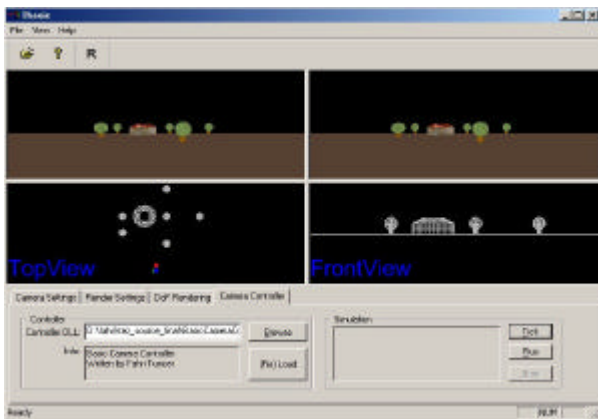


Figure 2 Screen shot from the software

4. Active Vision and Cognitive Map Construction Strategy

In this study, we aim to construct a world-centric grid-based cognitive map because of its simplicity and applicability. World-centric maps are represented in a global coordinate space. Since disambiguation of similar but different places is hard and extrapolation from individual measurements to measurements nearby is very difficult for robot-centric approaches, dominant approaches to date generate world-centric maps [16]. Occupancy grid mapping represents maps by fine grained grids that model the occupied and free space of the environment. Since metric maps are finer grained than topological, places that look alike can be disambiguated more easily.

Although this high resolution comes at a computational price, it helps to solve various hard problems.

In this study, 3D mapping is aimed and each grid codes a 1x1x1 unit cube of world unit. We assume that the agent's initial location in the world and initial orientation with respect to global North are known. We also assume that agent is aware of its internal camera

parameters such as gaze direction, focal length, field of view and base distance.

The agent centered depth information is obtained as in Section 2. Depth also provides a sort of obstacle location information in the field of view. This agent centered location information is converted to world centered location information by $z_W = R z_C$ where R is the rotation and translation matrix which converts the camera coordinates $z_C = (x_c, y_c, z_c)$ to world coordinates $z_W = (x_w, y_w, z_w)$. For the view shown in Figure 2, a multi-purpose user interface for displaying the calculated feature point information is shown as colored in Figure 3.

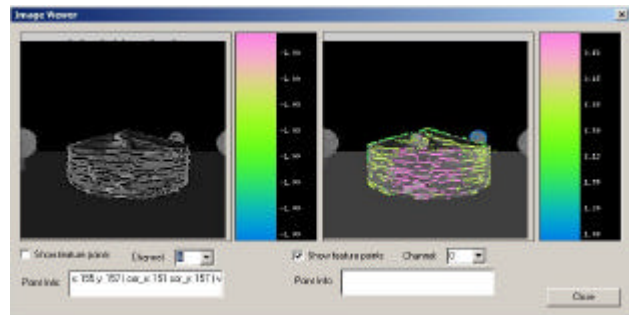


Figure 3 Interface for displaying feature point information. Locations of feature points for the house are shown in left camera view as black dots and disparities found are shown on the right camera view in colors.

With the pull-down menu called 'Channel' user can select to see camera centered coordinates, disparity, depth and world centered coordinates of feature points in both images. Also, if the user clicks on a feature point by mouse, all the information about that feature point is listed in the 'Point Info' dialog box. The image shown on the left side is the left camera view and feature point locations are shown on it for this time. The image on the right side is the right camera view and the disparity calculated for the right feature points is shown on it for this time.

In order to construct the cognitive map the following recursive active vision and exploration strategy is followed:

1. Initially, cognitive map grid values are set to zero, agent is located and oriented in the world and the cameras take the first stereo view, i.e. images are rendered.

2. The disparity and depth are calculated as in Section 2 and the world coordinates of the feature points are calculated as mentioned above. The resulting values for feature points could be observed by the interface as in Figure 3 if desired.

3. For the current view, map is updated. World centered location information for each feature point is

used to fill the cognitive map such that if a feature point belongs to a grid on the map, that grid's value is increased by 1.

4.If something new is observed in the view, the agent is moved in front of the nearest object to take a sharper view and steps 2 and 3 are repeated. Then the agent starts to turn around that object by looking at the object all the time and keeping a distance between itself and the object in the view. For each move while turning around the object, steps 2, 3 and 4 are repeated.

5.If nothing is observed, or if turning around the object is fullfilled, the agent makes a 90° counter-clock wise turn around itself. If a new object is observed then and steps starting from 2 are repeated. If a new object is not detected, the agent continues to make 90° counter-clock wise turns around itself. If nothing is observed until the agent makes a full turn around itself then the agent moves to a different location. In this move, the already constructed cognitive map is checked and the most unexplored direction is selected and the agent starts to move in that direction until a new object is seen or until the borders of the world are reached. If a new object is seen, steps starting from 2 are repeated. If nothing is observed then the agent starts to turn around itself and step 5 is repeated.

6.During all these movements the 3D cognitive map is updated gradually. In the end the 3D cognitive map of the environment is obtained. For example in Figure 4 the 3D featurepoint locations for a tree top (i.e. sphere) in the 3D cognitive map are shown. The occupancy information (i.e. top view) of the whole environment can be calculated by summing the grid values in y-axis as shown in Figure 5. This gives us the usual 2D environmental map. In this figure, the small disks are for the trees and the large disk is for the hause.

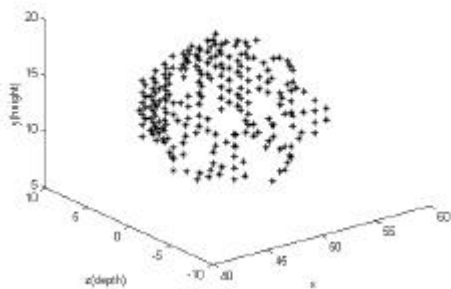


Figure 4 Feature point locations for a tree top (i.e. sphere) in the 3D cognitive map.

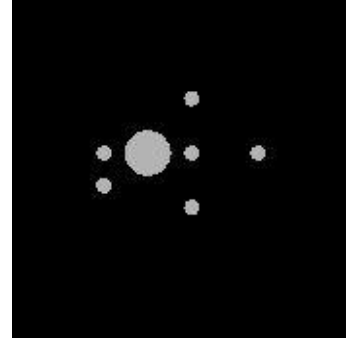


Figure 5 The occupancy (i.e. top view) of the environment obtained from 3D cognitive map by summing the grids in y axis.

5. Active Vision and Object Recognition

While exploring the environment for the purpose of map construction, when a new object is seen the agent gets close to that object and turns around it while looking at it and gets the 3D feature point locations of that object as shown in Figure 4. These locations are (x,y,z) triples. These are used in Equation (10) in order to understand the shape of the object. This equation is called the general quadratic equation in three variables [17, 18]. We assumed that the parameter A is non-zero, since the shapes in our environment is sphere, cylinder or cone. Then dividing the equation by A it becomes (11). Reorganizing the equation as in (12), the unknown equation parameters can be solved as in (13) where \tilde{V} is the pseudo inverse of V (14) and X is the vector in (15). Here N is the number of (x,y,z) triples.

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + 2Gx + 2Hy + 2Kz + L = 0 \quad (10)$$

$$x^2 + B'y^2 + C'z^2 + 2D'xy + 2E'xz + 2F'yz + 2G'x + 2H'y + 2K'z + L' = 0 \quad (11)$$

$$B'y^2 + C'z^2 + 2D'xy + 2E'xz + 2F'yz + 2G'x + 2H'y + 2K'z + L' = -x^2 \quad (12)$$

$$\begin{bmatrix} B' \\ C' \\ D' \\ E' \\ F' \\ G' \\ H' \\ K' \\ L' \end{bmatrix} = \tilde{V} * X \quad (13)$$

$$V = \begin{bmatrix} y_1^2 & z_1^2 & 2x_1y_1 & 2x_1z_1 & 2y_1z_1 & 2x_1 & 2y_1 & 2z_1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ y_N^2 & z_N^2 & 2x_Ny_N & 2x_Nz_N & 2y_Nz_N & 2x_N & 2y_N & 2z_N & 1 \end{bmatrix} \quad (14)$$

$$X = \begin{bmatrix} x_1^2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_N^2 \end{bmatrix} \quad (15)$$

After finding the parameters, we can understand if the shape is sphere or cylinder or cone by doing the following analysis. Let us define

$$S = \begin{bmatrix} 1 & D' & E' \\ D' & B' & F' \\ E' & F' & C' \end{bmatrix} \quad (16)$$

$$Q = \begin{bmatrix} G' \\ H' \\ K' \end{bmatrix} \quad (17)$$

Here S is a non-zero real symmetric matrix [18]. S has real characteristic values, $\lambda_1, \lambda_2, \lambda_3$, at least one of them is different than zero. Here since we deal with 3 basic shapes of sphere, cone and cylinder, we can say that all three are different than zero. The values of $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of the matrix S . Since we know that all three values are different than zero then the equation can be rewritten as in (18) where k is calculated as in (19) and (20) where u_1, u_2, u_3 are the eigenvectors of S .

$$I_1 \tilde{x}^2 + I_2 \tilde{y}^2 + I_3 \tilde{z}^2 = k \quad (18)$$

$$k = \frac{\overline{G'}^2}{I_1} + \frac{\overline{H'}^2}{I_2} + \frac{\overline{K'}^2}{I_3} - L' \quad (19)$$

$$\overline{G'} = Q^T u_1, \quad \overline{H'} = Q^T u_2, \quad \overline{K'} = Q^T u_3 \quad (20)$$

When we calculate $a = \sqrt{\frac{k}{I_1}}$, $b = \sqrt{\frac{k}{I_2}}$, $c = \sqrt{\frac{k}{I_3}}$ we obtain three nonzero values and these give us the information about the shape of the observed object. If the object is a sphere (for example the top of the trees) then

these numbers are equal and give us the radius of the sphere. If the object is a cylinder (for example the body of the trees or the house) then two of these numbers are equal and the other one is very big and different from the others. If the object is a cone (for example the roof of the house) then one of these numbers is imaginary (i.e. square of the number is less than zero) and the other two are greater than zero.

6. Conclusion

We constructed a virtual environment as a platform easier than the real robotic applications in order to develop an algorithm for the purpose of environmental map construction and object recognition from stereo image data only. The environment to be processed in this study is a structured environment but the main target is to explore an unstructured environment in which there are no predefined structures such as corridors, walls or objects. An extreme is having any type of structure in the environment after a natural disaster or a terrorist attack. There are many real time map construction for structured environments but for unstructured environments mapping needs much more effort and still needs more research.

In this study we use image in order to acquire the map. No odometry information is supplied. Normally the robot is subject to cumulative noise and internal odometry can not be used reliably over long periods of time. But in this study, we use images and especially stereo as sensors which helps in compensating the odometry error. Since the robot sees where to go, assuming that there are enough cues in the environment the robot can correct its movement error. But this odometry error correction is not the scope of this study. In the future studies, some random odometry error will be assumed and correction of the trajectory using stereo image information will be investigated. In this study, we assume that there is no odometric error.

We developed an intelligent map construction algorithm which uses stereo data only. Since stereo algorithms are time consuming, usually very simple and fast ones are preferred in robotic applications but they are not very sufficient. Our stereo algorithm takes 5 seconds at a Pentium II computer with 256 MB RAM. We have presented a stereo correspondence algorithm where steerable filters are used as feature point detecting method. The stimulus orientation found using steerable filters is very accurate given the small number of filters used. Although the feature points are sparse, since they are the points of high contrast edges including the bounding contours of objects, they still prove to be informative. Correspondences between similarly oriented feature points are located using the phase information. Phase is robust to small scale differences. Unfortunately, there are image locations where phase is singular and can

not be reliably used. In this study, by performing phase comparisons at multiple scales and by using magnitude as a weight in phase similarity check we overcome these difficulties. With better and better stereo algorithms, and with the belief that robot should act as human beings, we can anticipate that stereo vision will be one of the most widely used sensors for mobile robots. Thus our research in order to increase the speed of our algorithms is still continuing.

We introduce an object recognition algorithm from feature point (edge) information in 3D world. In doing this general quadratic equation is used. In this study the objects are classified only in three main groups: sphere, cylinder or cone. However, this general quadratic equation will be used in the future studies in order to make classifications in many other shapes.

7. References

- [1]. B. Jerbic, K. Grolinger, B. Vranjes, "Autonomous agent based on reinforcement learning and adaptive shadowed network", *Artificial Intelligence in Engineering* 13, 1999, pp. 141-157.
- [2]. R. Ramloll, D. Mowat, "Way finding in virtual environments using an interactive spatial cognitive map", *Proceedings of Fifth International Conference on Information Visualisation*, 2001, pp. 574 -583.
- [3]. D. Terzopoulos, T. F. Rabe, "Animat vision: Active vision in artificial animals", *Proceedings of Fifth International Conference on Computer Vision*, 20-23 Jun 1995, pp. 801 -808.
- [4]. H. Voicu, N. Schmayuk, "Three dimensional cognitive mapping with a neural network", *Robotic and Autonomous Systems* 35, 2000, pp. 23-36.
- [5]. M. Quoy, P. Laroque, P. Gaussier, "Learning and motivational couplings promote smarter behaviours of an animat in an unknown world", *Robotics and Autonomous Systems* 38, 2002, pp. 149-156.
- [6]. G. Li, B. Svensson, "Navigation with a focus directed mapping network", *Autonomous Robots* 7, 1999, pp. 9-30.
- [7]. M. O. Franz, B. Schölkopf, H. A. Mallot, H. H. Bülthoff, "Leaning View Graphs for Robot Navigation", *Autonomous Robots*, 5, 1998, pp. 111-125.
- [8]. A. Elfes, "Sonar-based real-world mapping and navigation.", *IEEE Journal of Robotics and Automation*, RA-3(3):249-265, June 1987.
- [9]. S. Thrun, "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation", *Artificial Intelligence*, 99 (1), 1998, pp. 21 -71.
- [10]. D. Murray, J. Little, "Using real-time stereo vision for mobile robot navigation", *Proceedings of the IEEE Workshop on Perception for Mobile Agents*, Santa Barbara, CA, June 1998.
- [11]. A. Erol, "Automatic Fingerprint Recognition", *Ph. D. Thesis*, METU, Ankara, 2001.
- [12]. D. Jepson, D. J. Fleet, "Phase singularities in scale space", *Image and Vision Computing*, vol. 9, no. 5, 1991, pp. 338-343.
- [13]. T. D. Sanger, "Stereo disparity computation using Gabor filters", *Biol. Cybern.*, 59, 1988, pp. 405-418.
- [14]. DeAngelis, "Seeing in three dimension: the neurophysiology of stereopsis", *Trends in Cognitive Science*, vol. 4, no. 3, 2000, pp. 80-89.
- [15]. Tunçer, F., *Image Synthesis for Depth Estimation Using Stereo and Focus Analysis. M.Sc. Thesis, Department of Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey, 2002.*
- [16]. S. Thrun, *Robotic Mapping: A Survey, Technical Report CMU-CS-02-111*, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.
- [17]. H. I.Karakas, *Analytic Geometry*.
- [18]. D. F. Rogers, J. A. Adams, *Mathematical Elements for Computer Graphics*, Second Edition, McGraw-Hill International Editions.