VISION-TACTION INTEGRATION FOR SURFACE REPRESENTATION

Nader I. Rafla and Francis L. Merat

Center for Automation and Intelligent Systems Research
Department of Electrical Engineering and Applied Physics
Case Western Reserve University
Cleveland, Ohio 44106

ABSTRACT

Object description and representation are important for applications such as intelligent robotics or inspection where sensor information is reduced to symbolic form for interpretation. Because of the nature of the sensor data, objects are often described in terms of connected surfaces. We describe a new method called Vision-Taction Exploration (VTE) for generating surface descriptions from range vision and tactile sensor data. The range vision systems available provide sparse 3D data about surfaces. This data is partially processed to provide us with primary surface features such as surface points and surface normals. With the use of tactile and force-torque sensors under position control, supplementary data is obtained and processed to generate surface features identical to those generated from the vision data. These two sets of data are integrated and processed in a higher level for surface representation and classification. The VTE process levels are described. The touch system has been implemented and the resulting surface description is also presented.

1. INTRODUCTION

Conventionally, surface representation is usually performed using either visual data [1,2,3], or tactile data [4,5,6]. With the advent of touch sensors [7,8,9], we propose in this paper a new method to generate surface descriptions by processing visual range image and touch data as one set of data.

The processing occurs at two different levels, point and kernel. At the point level, vision and touch data are separately processed to estimate the surface normal at each data point and the results of each sensor are integrated into a single set of data that is passed to the kernel level where surface descriptions are generated.

The touch system consisted of a robot manipulator arm and a Lord Corporation two-finger gripper; each finger contained a tactile array sensor and a force-torque sensor underneath the array. Surface probing was done using a mechanical probe (aligned with the manipulator tool z-axis) grasped in the Lord gripper. When this probe contacts the surface to be explored, the tactile array and torque data are analyzed to position the pointer normal to the surface. The probe direction then gives the surface normal at the point of contact while the probe tip position indicates the 3D position of the point. This surface data is processed to generate a surface description identical to those generated from vision data.

At the kernel level, a starting point is used to generate a surface candidate. The correlation between the surface normals of the point being classified and previously classified adjacent surface points, is used to determine if the candidate point belongs to the given surface. When all surface points have been classified, the individual surface elements are further classified by type as planar, cylindrical or spherical. This classification is based on the mean and standard deviation of the surface normal within that surface. If these statistics are zero, the surface is planar; otherwise, it is curved. For curved surfaces, the existence of a central axis is detected and the surface is then classified as cylindrical or spherical.

2. The POINT LEVEL

At the point level, vision data is processed to estimate the surface normal at each sampled point. Touch data derived from the touch system is processed differently to estimate the same kind of point information. Both sets of data are integrated into a single set of data that is passed to the kernel level.

A) Vision Image Processing

A point extraction algorithm that scans the range image points and provide an estimation about the surface normal at each extracted point that contains 3D information is shown in figure 1.

Input: Range vision image **Output:** Array of sampled points with their normal direction
1. Find background points 2. mask out background areas 3. Scan sampled points in both directions 4. Calculate the surface normal at each point 5. Form the output array

Figure 1 - Point Algorithm

The algorithm scans the range image points and masks out background areas. These areas are defined as areas enclosed by a rectangle whose corner points are background points (background points are those points which do not belong to the object and are known, in our case, by calibrating the vision sensor with no object present.) The constraint used to extract these areas is that the rectangle area be less than a preset value. For each sampled point, The surface normal $N(i,j)$ at the point (i,j) is calculated using the equation [10]

$$N(i,j) = \frac{v_i(i,j) \times v_j(i,j)}{|v_i(i,j) \times v_j(i,j)|}$$

where $v_i(i,j)$ and $v_j(i,j)$ are the partial derivative vectors at (i,j) with respect to i and j respectively and (x) denotes the vector cross product. These partial derivatives are approximated by a finite difference operation by convolving the surface points with the 3 by 3 matrices given below

$$v_i = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad ; \quad v_j = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

B. Touch Data Processing

The touch system consists of a robot manipulator arm and a lord corporation gripper; each finger contains a tactile array sensor and a force-torque sensor underneath the array. A mechanical prob, aligned with the manipulator z axis, is grasped in the Lord gripper. The system is shown in figure 2.

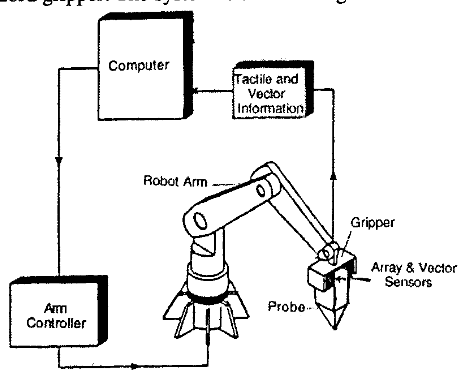


Figure 2 - Touch System Configuration

Before the touch data processing system can be described, we must define some notation that geometrically describes the system configuration. Each of the tactile sensors generates an image array of sensor element mechanical deflections. Let I_R denote the right hand side array sensor as viewed from the gripper wrist, and I_L denote the left hand side array sensor. Each array contains 10x16 elements representing the deflection of the touch surface. The two vector sensors provide a 1x6 vector array representing the forces and moments along and about the three coordinate axis of the touch sensor surface. Similar to the image arrays let the vector arrays be designated V_R and V_L. O_R be the origin of the coordinate system for I_R and V_R and is chosen to be the center element of I_R. The origin O_L for I_L and V_L is similarly chosen. The vector connecting the two origins O_L and O_R in the direction from L to R is considered the x-axis of the touch coordinate system. The middle point of this line in the coordinate center "O". The direction of the vector originating at "O" and parallel to the short tactile surface is considered the y-axis of the touch coordinate system. The remaining perpendicular direction to both x and y vectors is the z-axis of the touch coordinate system. This is the same as both the tool z-axis of the manipulator and the vision data. Figure 3 shows the geometric structure described above.

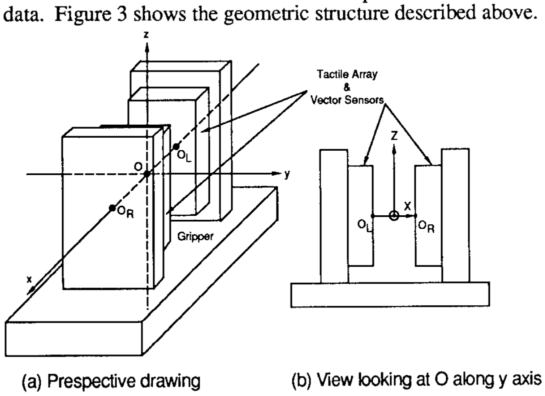


Figure 3 - Geometric structure of the touch system

C. Touch Data Analysis

The primary analysis of the touch data is in the extraction of the surface normal of a point under contact. Since the tactile images I_L and I_R are provided when the tactile sensors are touching a surface, the initial values of these arrays are determined for the gripper grasping the probe with no surface contact. When the probe tip is in contact with a surface, the new tactile images are subtracted from the initial values, some local averaging is done to reduce image noise, and analyzed to determine if surface contact has been established. The constraint used in this process is that the absolute value of the differences between left and right tactile image arrays is greater than an experimentally determined value. Next, the vector data V_R and V_L is analyzed and compared in the x and y directions to determine that the prob is perpendicular to the surface. The criteria used is as follows:

* Let M_x and M_y be the absolute value of the differences between the left and right vector array moment elements in the x and y directions respectively.
* If $M_x > M_y$ motion is derived in the x direction otherwise in the y direction. The positive and negative direction is detected by comparing the corresponding force elements in both V_R and V_L.
* If the difference between M_x and M_y is less than a threshold value, the probe is in the direction of the surface normal at that point and the process is repeated for another contact point.

A flow chart of the touch data processing system in shown in figure 4.

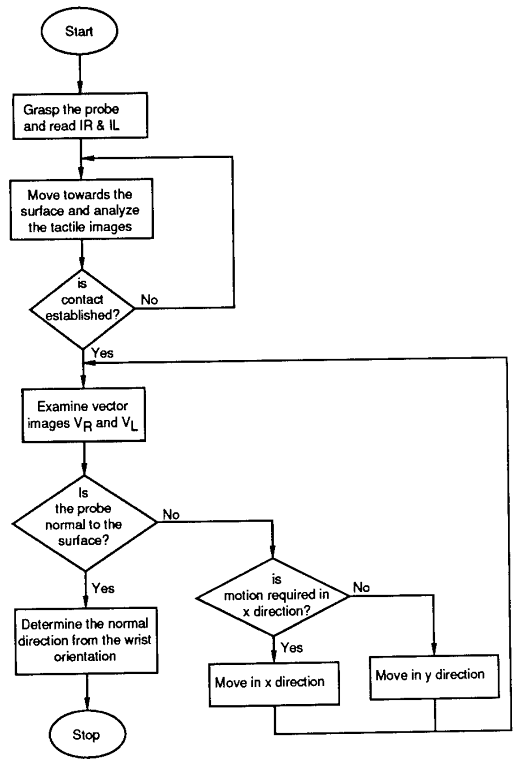


Figure 4 - Flow chart for touch data processing system

3. KERNEL LEVEL PROCESSING

At the kernel level, the points extracted by the touch point level process are checked with the points extracted from the vision point level process for similarity using the correlation between the dot products of the surface normals at these points. This method does not require either the Gaussian or mean curvature calculations nor a periori knowledge about the surface. A central (in the center of the surface) touch point located at (x,y) is chosen to initiate the process. The mean m_C and standard deviation s_C of the dot product of the surface normal (N_{xy}) at that point and other normals (N_{ij}) of vision data located within a window of width w surrounding the point can be calculated as:

$$\mu_C = \frac{1}{w^2} \sum_{i=x-(w-1)/2}^{x+(w-1)/2} \sum_{j=x-(w-1)/2}^{x+(w-1)/2} N_{ij} \cdot N_{xy}$$

$$\sigma_C = \left(\frac{1}{w^2} \sum_{i=x-(w-1)/2}^{x+(w-1)/2} \sum_{j=x-(w-1)/2}^{x+(w-1)/2} \left(N_{ij} \cdot N_{xy}\right)^2 - \mu_C^2 \right)^{1/2}$$

Once these properties are calculated, the correlation of dot products of surface normals can be corrected with the normal determined by the touch sensor. It's value determines if two points do belong to the same surface by comparing it with a pre-experimental determined threshold. The point belongs to the surface if its correlation is greater than the threshold. The process is repeated for other points until all points are examined. The algorithm used is shown in figure 5.

Input: Surface points extracted from vision and touch data **Output:** Surface Kernels
1. Choose a touch point 2. Locate vision points inside a window centered at the touch point and of width w 3. Find the correlation of dot product of surface points, within the window, and the touch point 4. Determine if they belong to the same surface 5. Repeat for another touch point

Figure 5. Kernel level algorithm

4. SURFACE CLASSIFICATION

After the surface kernels are extracted, they are classified as planar, cylindrical or spherical. This classification is based on the mean and standard deviation of the unit vector from a point to a neighboring point and the normal at that point. When the surface is planar these statistics are zero, otherwise the surface is curved. Therefore, these statistics were used to distinguish a planar surface from a curved one. The difference between a cylindrical surface and a spherical one is the existence of a central axis for a cylindrical surfaces. The axis is perpendicular to the normal at every point in the surface. If the normal at a point is denoted as N_i and the central axis direction denoted as C then from [10] the error of the central axis is calculated and defined by the following equation:

$$e = \sum_{i=1}^{n} \left(C \cdot N_i\right)^2$$

This error is minimum if C is a central axis. The directional unit vector C is calculated by minimizing that error equation. The condition is satisfied if $\nabla (e + \lambda C \cdot C) = 0$
This condition can be rewritten as follows:

$\sum_{i=1}^{n} \left(N_i^T \cdot N_i\right) C + \lambda C = 0$ and the solution for c is the eigenvector of the matrix $\sum_{i=0}^{n} \left(N_i^T \cdot N_i\right)$ with minimum eigenvalue which is the error of the vector C. If the error is less than some threshold value, the surface is cylindrical. Figure 6 shows the flow chart for the surface classification algorithm.

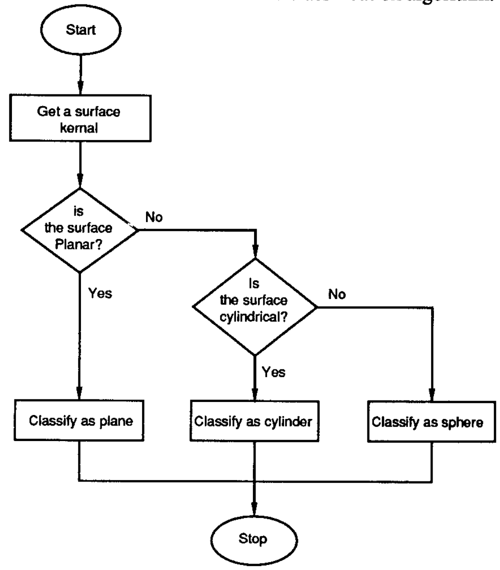


Figure 6 - Classification system flow chart

5. EXPERIMENTAL RESULTS

Experiments have been done to test the (VTE) algorithm described in this paper using 64x64 synthetic surface range images with added Gaussian noise with maximal error of 1 mm and touch data acquired from the same surfaces. The surface images are shown in figure 7. The threshold used in these experiments was 0.9. Figure 7.a, shows a synthesized range image which contains a plane surface 5cm in length and 3cm in width. Figure 8 shows the surface kernel using a touch point at (32,32). The experiment is repeated for a cylindrical surface with radius 3 cm and a touch point at (32,13). The image and the results are shown in figure 7.b and 9. An image for a sphere with a radius of 5 cm is shown in figure 7.c. The results after using the (VTE) algorithm are shown in figure 10. The touch point position used for this surface was (35,40).

The VTE algorithm performed well in all three cases. In all three cases, few points within the classified kernel were not correctly identified as belonging to the surface kernel. This is due to the added Gaussian noise. Since our goal is to use these surface kernels as the starting points for more comprehensive surface segmentation algorithms and model based object recognition system, these errors are tolerable. All the points were correctly identified with no added noise.

6. CONCLUSIONS

We have demonstrated the feasibility of a system that utilizes a combination of visual information and touch data in the representation and classification of surfaces. An algorithm (VTE) has been implemented which succeeded in accurately representing and classifying any kind of the three primitive surfaces: planar, cylindrical, or spherical. The results show that the (VTE) algorithm could be used to develop object models from multisensor data.

7. REFERENCES

[1] M. Potmesil, "Generation of 3D Surface Descriptions from Range Images of Pattern Illuminated Objects," Proceedings of the IEEE Pattern Recognition and Image Processing Conference, 553-559, 1979

[2] T. J. Fan, G. Medioni, and R. Nevatia, "Description of Surfaces from Range Data Using Curvature Properties," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 86-91, June 1986

[3] Hsianglung Wu and Francis Merat, "Generating Object Descriptions from Range Data Using Feature Extraction by Demands," IEEE International Conference on Robotics and Automation, Vol. 2, 942-946, 1987

[4] R. E. Ellis, "Extraction of Tactile Features by Passive and Active Sensing," Proceedings of the International Conference of Intelligent Robotics and Computer Vision, 289-294, 1984

[5] Michael K. Brown, "The Extraction of Curved Surface Features with Generic Range Sensors," International Journal of Robotics Research, Vol. 5, No. 1, 1985

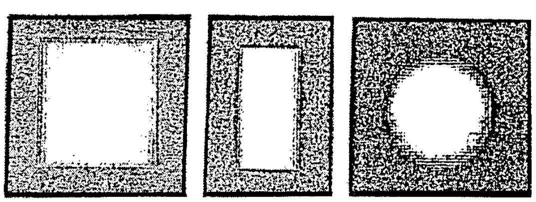
[6] W. Eric, L. Grimson and T. Lozano-Perez, "Model-Based Recognition and Localization from Tactile Data," International Journal of Robotics Research, Vol. 3, No. 3, 3-25, 1984

[7] R. C. Luo, F. Wang and Y. Liu, "An Imaging Tactile Sensor with Magnetostrctive Trandedustion," Proceedings of the International Conf. of Intelligent Robotics and Computer Vision, Cambridge Ma., Nov. 1984.

[8] B. E. Robertson, A. J. Walkden, "Tactile Sensor Systems for Robotics," Proceedings of the International Conf. of Intelligent Robotics and Computer Vision, Cambridge, MA, Nov. 1983.

[9] M. Briot, "The Utilization of an 'Artificial Skin' Sensor for the identification of Solid Objects," Proceedings of the 9th International Symposium on Industrial Robots, Washington, D.C., 1979.

[10] Manfredo P. do Carmo, "Differential Geometry of Curves and Surfaces," Prentice-Hall, 1976.



(a) Planar (b) Cylindrical (c) Spherical

Figure 7 - Surface images with added maximal Gaussian error of 1 mm

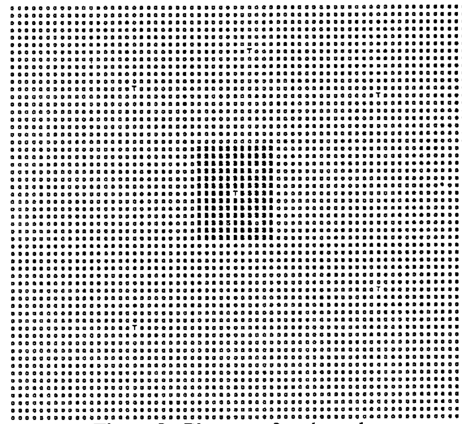


Figure 8 - Planar surface kernel using touch point at (32,32)
T = Touch points S = Surface points

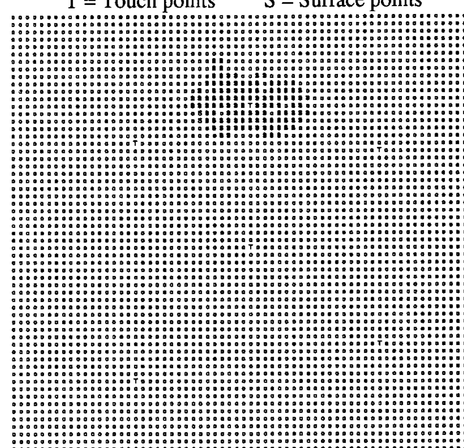


Figure 9 - Cylindrical surface kernel using touch point at (32,13)
T = Touch points S = Surface points

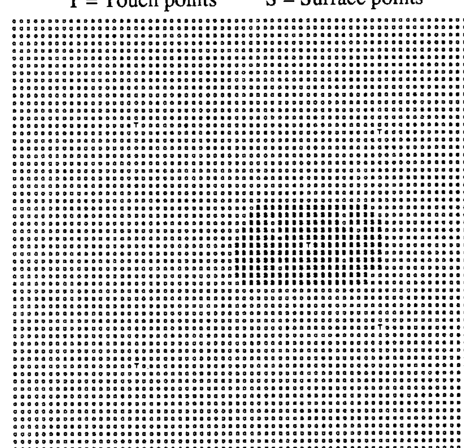


Figure 10 - Spherical surface kernel using touch point at (35,40)
T = Touch points S = Surface points