Seventh International Conference on Control, Automation, Robotics And Vision (ICARCV'02), Dec 2002, Singapore

Vision-Based Non-Smooth Kinematic Stratified Object Manipulation

Yejun Wei and Bill Goodwine
ywei@nd.edu and bill@controls.ame.nd.edu
Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Abstract— **This paper presents experimental demonstration and verification of a vision-based stratified motion planning method for the case where the base manifold upon which the motion planning occurs is not smooth. Robotic applications of the method include motion planning for legged robotics over non-smooth (but known) terrain and manipulation of non-smooth objects with multiple robotic manipulators. Experimental results with multiple robots manipulating a common non-smooth object is presented.**

1 Introduction

This paper presents experimental robotic manipulation results, which demonstrate and verify a motion planning algorithm for systems on non-smooth domains. As schematically illustrated in Figure 1, we consider the manipulation planning problem for four fingers manipulating a non-smooth object. The theoretical foundation for these experiments was presented previously by the authors in [1]. The method is an extension of a novel control strategy which considers motion planning for robotic systems which are characterized by switching dynamics. The basis for this work was the development of a "stratified motion planning" algorithm which provided a means for motion planning for smooth systems with switching dynamics [2], [3], [4], [5], [6], [7], [8]. Experimental results for smooth systems were presented in [9], [10].

Specifically, the previous work of the authors assumed that the configuration manifold for the system under consideration was *smooth* and that the discontinuous nature of the dynamics of the system resulted only from the intermittent physical contact among various elements of the overall system. A consequence of this assumption was that a particular critical element (the "bottom stratum," described subsequently) was smooth. Previous results did not consider the case where the bottom stratum was not smooth, which is a case that includes legged locomotion over non-smooth terrain and manipulation of non-smooth objects, which is the focus of this paper.

The main difficulty with such systems, and stratified systems in general, is to determine a method to analytically incorporate, either in an analysis tool or control synthesis algorithm, the discontinuous nature of the equations of motion for the system. Incorporating the discontinuities of the equations of motion of

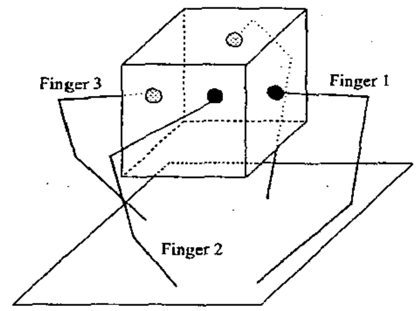

Figure 1. Non-smooth object manipulation.

a system into a general motion planning algorithm is difficult because almost all motion planning methods assume that the equations of motion are smooth.

Many efforts considered the analysis of grasp stability and force closure [11], [12], [13], motion planning assuming continuous contact [14], [15], [16] and haptic interfaces [17], [18], [19]. Finger gaiting has been implemented in certain instances [20], [21], [22] and also partially considered theoretically [23], [24], [25]. Perhaps the approach which most closely mirrors that of the subject of this proposal is in [26] where notions of controllability and observability from "standard" control theory are applied to grasping (however, these results are limited to the linear case and do not allow for fingers to intermittently contact the object). In contrast to the current work, none of these methods directly use the inherent geometry of stratified configuration spaces to formulate results which span many different morphologies and assumptions.

From an experimental perspective, the main shortcoming of the theory is that the control strategy is fundamentally open loop. We therefore incorporate, in a manner detailed subsequently, a vision based robotic control method to provide periodic updates of the system configuration so that the open loop planning can be appropriately periodically modified. Most implementations of computer vision entail calibration of the cameras and calibration of the kinematics of the robots. If both the robots and cameras are accurately calibrated, this method could provide position and orientation information of the manipulated object. However, historically problems with three-dimensional implementation of such a strategy have centered on difficulties with achieving and sustaining accuracy of the calibration of both the robots and cameras [27],[28],[29]. A second approach to vision-based robotic control is visual servoing [30],[31]. However this approach is problematic for our problem due to the nonholonomic nature of the relationship between the motion of each robot and the response of the manipulated object. The approach we utilize is Camera-space manipulation, the details of which are presented in Section 3.

2 Non-Smooth Stratified Systems

This work is an extension of previous work by the authors; therefore, a short review of previous results is necessary. This discussion omits many of the necessary mathematical details, and the interested reader is referred to [5], [2], [3], [9], [1], [4], [6], [7], [8] for a complete, detailed exposition.

Consider the case of the four fingers manipulating the cube illustrated in Figure 1. If each finger has, say, three revolute joints, then the overall configuration space for the system is $S_0 = \mathrm{SE}(3) \times S^{3\times 4}$, where SE(3) describes the configuration of the cube and $S^{3\times 4}$ represents the configuration of the joints. If the object were smooth, then the set of all configurations where one finger contacts the object defines a *smooth* codimension one submanifold of S_0. However, since the object is not smooth, the set of configurations where the finger contacts the cube will be the *union* of 6 disconnected smooth manifolds *with boundary*.

In general, we use stratum $S_{I_m J_n K_p L_q}$, where $1 \leq I < J < K < L \leq 4$ and m, n, p and q are different integers between 1 and 6, to represent the configuration when four fingers I, J, K and L are in contact with surfaces m, n, p and q respectively. Similarly, stratum $S_{I_m J_n K_p}$ represents the configuration when three of the four fingers, I, J and K, are in contact with the surfaces m, n and p of the object respectively. The level of the stratum is referred to as its codimension. Thus, the bottom stratum for the structure our system is on the 4th level and contains all the strata with codimension 4 representing all the four fingers are in contact with the cube. Similarly, all the strata representing three of the four fingers are in contact with the cube are in level 3. Part of the combinatorial structure of the stratified system is shown in Figure 2, where the nodes of the figure represents different strata, the edges connecting the nodes indicate that it is possible for the system to move from one stratum to another. Thus, if the nodes are connected by an edge, the system can move between the strata, if there is no edges between two nodes, the system cannot move between them directly.

Comparing the structure of this stratified system

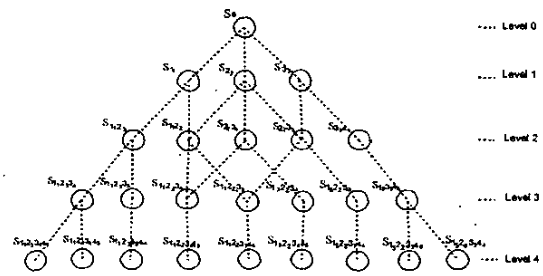

Figure 2. The Structure of the Non-Smooth Stratified System.

with the system of robot fingers engaging a smooth object, we can find that for the stratified system with multiple fingers to manipulate a smooth object, there is just one bottom stratum in the configuration structure. However, for our stratified system for four fingers to manipulate a cube object, the bottom stratum is the union of $P_6^4 = 360$ manifolds with boundary. In the rest of the development, we will consider each of these manifolds with boundary as a separate stratum.

Although we have presented the structure of a non-smooth stratified system by way of a particular example, namely, four fingers manipulating a cube, formulating the generic structure is straight forward. In particular, a stratum is simply a smooth manifold with boundary and a single, smooth bottom stratum does not necessarily exist, but is, in fact, the union of multiple strata.

Having developed the structure of a non-smooth stratified system, we consider motion planning for non-smooth systems, which only involves one additional complication relative to the smooth stratified system case. In particular, since the bottom stratum is not simply a smooth manifold, then the nominal trajectory will not be contained within a single stratum. To see this, consider the same example of four fingers manipulating a cube and consider an ending configuration wherein the four fingers are in contact with different faces of the cube than upon which they started. Since the nominal trajectory is computed in the bottom stratum (all the fingers in contact with the cube), the nominal trajectory will need to switch among the various strata from which the bottom stratum is comprised.

3 Camera-Space Manipulation

Camera-space manipulation is utilized in this experiment to circumvent the difficulties of calibrating the robot kinematics, the kinematics relationship among multiple robots, as well as the camera(s) to the very highly accurate degree necessary ([32],[33]). Each camera can form a projection from the 3D Cartesian (physical) space in view onto the cameras' 2D image plane as shown in Figure 3.

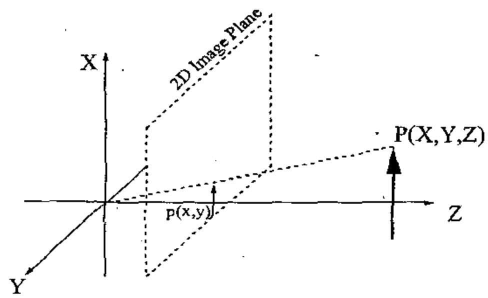

Figure 3. Mapping from the Cartesian coordinates to the image plane coordinates.

The camera model used here is a perspective projection model. As shown in Figure 3, the $x-$ and $y-$axes form a basis for the image plane, the $z-$axis is perpendicular to the image plane (along with the optic axis), and with the origin located at distance f behind the image plane, where f is the focal length of the camera lens. The perspective projection model can be described by

$$x_c = f\frac{X}{Z} \text{ and } y_c = f\frac{Y}{Z},$$

where x_c and y_c are the image plane, *i.e.*, camera-space, coordinates of the point (X, Y, Z).

This projection is a surjective mapping where each point on the image plane corresponds to a ray in 3D space. An approximation model, "orthographic camera model," is introduced in ([34],[35]). Thus, given a physical point on the robot manipulator in the view range of a camera, its image position in that camera's 2D image plane can be determined by

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = F(X, Y, Z; \vec{C}), \qquad (1)$$

where, F is the mapping from $3D$ physical space to $2D$ image plane and $\vec{C}$ is a ***visual vector***, which includes 6 view parameters, *i.e.*, $\vec{C} = [C_1, C_2, C_3, C_4, C_5, C_6]^T$, used to identify the local relationship between robot joint configuration and the camera-space location of the points on the manipulator. The detailed description of the mapping F and $\vec{C}$ is shown in ([36],[32],[37]), where,

$$\begin{aligned} x_c &= (C_1^2 + C_2^2 - C_3^2 - C_4^2)X + 2(C_2C_3 + C_1C_4)Y \\ &\quad +2(C_2C_4 - C_1C_3)Z + C_5 \equiv F_x(X, Y, Z; \vec{C}) \\ y_c &= 2(C_2C_3 - C_1C_4)X + (C_1^2 - C_2^2 + C_3^2 - C_4^2)Y \\ &\quad +2(C_3C_4 + C_1C_2)Z + C_6 \equiv F_y(X, Y, Z; \vec{C}). \end{aligned}$$

The idea behind the method is to exploit a nominal robotic kinematics model as well as the fact that both target object and manipulator can be seen in multiple, redundant camera views.

For one of the cameras, the view parameters $\vec{C}$ can be estimated through the acquisition of a large number of simultaneous image plane and physical space samples by minimizing:

$$\begin{aligned} J(\vec{C}) = \sum_{i=1}^{n_p}\Big[\sum_{j=1}^{n_{ci}} \Big\{ &(x_{c_{ij}} - F_x(X_i, Y_i, Z_i; \vec{C}))^2 \\ &+(y_{c_{ij}} - F_y(X_i, Y_i, Z_i; \vec{C}))^2 \Big\} W_j \Big] W_i, \end{aligned} \qquad (2)$$

where, n_p is the number of poses of robot joint rotation samples, n_{c_i} is the number of known points identified in the camera sample in the $i-th$ pose, W_i is a weight associated with the $i-th$ pose, and W_j is a weight given to each identified point in the image.

The above algorithm is based on the orthographic camera model, which requires that all the interested physical cues are close to each other. A process is introduced in [34], which eliminates this constraint and gives more accurately fitted view parameters.

Once the view parameters $\vec{C}$ have been determined, for each cue in a joint configuration, its location in the camera space can be determined by using equation (1). That mapping F is a surjective mapping. However, if two or more cameras are used to detect the common cues on the manipulator, there exists a bijective mapping G from the physical space to the image plane:

$$\begin{bmatrix} x_{c_{i_1}} \\ y_{c_{i_1}} \\ \vdots \\ x_{c_{i_m}} \\ y_{c_{i_m}} \end{bmatrix} = G(X_i, Y_i, Z_i, \vec{C}_1, \cdots, \vec{C}_m), \qquad (3)$$

where the cue (X_i, Y_i, Z_i) can be detected at the same time in m $(m \geq 2)$ cameras; $\vec{C}_i$ is the ith camera's ***visual vector***; (x_{c_i}, y_{c_i}) is the image point in the ith camera.

4 Vision-Based Manipulation

The flow chart illustrated in Figure 4 shows the manner in which CSM is incorporated into the stratified robotics finger gaiting algorithm. First, the robot moves along a "preplanned" trajectory during which image information regarding the location of cues placed on the end effector of the manipulators is acquired. Although there are multiple cues on each robot, each also has a unique black cue. From the image locations of the unique black cue in each pose an initial set of view parameters $\vec{C}$ can be determined by minimizing Equation 2. The non-unique white cues on the manipulator can be distinguished with the initial $\vec{C}$, and are used with the black cues to refine the view parameters. Once the view parameters is determined, the pose of the target object can be determined by determining the physical coordinates of the cues attached to the object by Equation 3.

Next, a trajectory is planned for one end effector to approach the target object, and this trajectory is divided into 5 subtrajectories. As the manipulator moves along the subtrajectories to approach the target object, the view parameters are updated based on images acquired when the end effector is at the end point of each subtrajectory. Once all the end effectors are in contact with the object finger gaiting can be achieved with each robotic finger following the trajectory produced from the the stratified motion planning finger gaiting algorithm. Periodically, images are acquired during the finger gaiting, and if the object configuration and contact coordinates are substantially different from the configuration and contact coordinates assumed by the open-loop stratified motion planning algorithm, a new open-loop trajectory is computed to reflect the updated configuration of the object and contact coordinates.

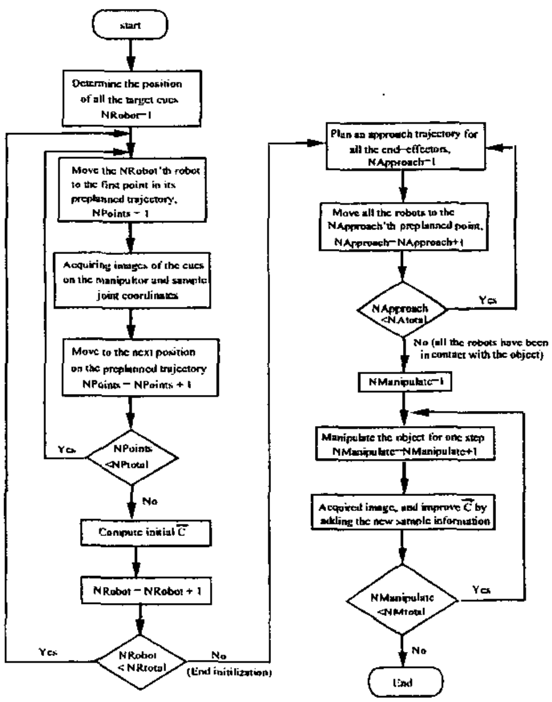

Figure 4. Flow chart representing the experiment procedure.

5 Experimental Implementation

An experimental platform has been developed to demonstrate the application of the algorithm outlined in Figure 4. Four Puma 560 robots are mounted on one common platform, where each robot is used to simulate a finger with six joints in the experiment. Three Galil motion control boards, each of which can control up to 8 axes, are installed on one Pentium IV 1.7 GHz PC running Linux operating system to control the motion of the 24 joints of the 4 robots. Pictures of the experimental platform are illustrated in Figures 5-10.

Three standard black and white Sony XC75CE cameras are used in the the experiment, each with a focal length of 25mm. Two cameras are mounted about 120 inches away from the target on the wall, with a separation of 100 degrees between them, and the other camera is mounted on the ceiling approximately 60 inches up from the robots. Two Picport Stereo image capture boards are installed on the PC, and all the image processing and trajectory planning is done on it.

The method is general in that it can accommodate any non-smooth terrain or object[1], in this paper we will primarily consider a cube object as the common object for the four fingers to cooperatively manipulate. A unique black cue and 3 white cues, with each cue at a known position relative to the cube center, are attached to the top surface of the cube. Once the view parameters are initialized, the cue images in the image plane can be distinguished and matched to the cues on the cube object and the manipulator, and the position and orientation of the cube object can thus be determined.

By moving each manipulator along a preplanned trajectory, and acquiring a series of simultaneous joint rotation and cue image views (32 in this experiment), we can initialize sets of view parameter $\vec{C}$ between the manipulators and cameras, where there is one $\vec{C}$ between each camera and manipulator, totally amounting to 12 sets of $\vec{C}$ with 3 cameras and 4 robots used in this experiment. Then a trajectory is planned for each manipulator to approach the object, and the $\vec{C}$'s are further updated during this approach process. Here, we employ a straight line connecting the current position of the manipulator and the specified contact point on the object as the preplanned trajectory, and this trajectory is divided into 5 subdirectories, where the manipulator is 4,2,1,0.3 and 0 inches away from the object along the line. Image samples are acquired while the manipulator is at the endpoint of each subtrajectory, and $\vec{C}$'s are again updated. Once the manipulators are engaged with the object, the robots begin to manipulate the object according to the stratified motion planning algorithm. Here, the goal motion is to rotate the cube about an axis oriented in the z di-

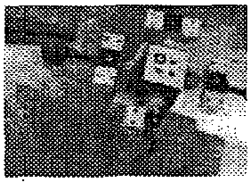

Figure 5. $\theta = 0.32$ rad

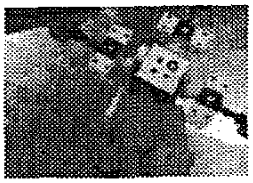

Figure 6. $\theta = 4.74$ rad

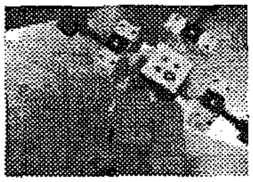

Figure 7. $\theta = 3.10$ rad

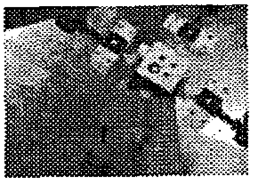

Figure 8. $\theta = 1.49$ rad

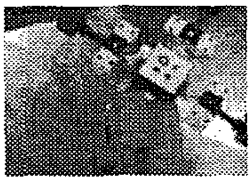

Figure 9. $\theta = 6.27$ rad

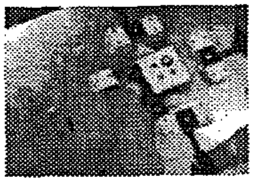

Figure 10. $\theta = 5.99$ rad

rection with an angular velocity $\{0\ 01\}^T$ radian/s for 7.308 radians, where, on the platform, the z axis goes along the positive vertical direction, and the x axis goes through the rightmost robot to the cube. After each step of motion, samples are again acquired, and the $\bar{C}$'s are correspondingly updated. The result for the above motion is shown in the next section.

6 Experimental Results

The pose of the cube as well as the manipulators during motion are shown in the Figures 5 through 10 wherein the cube is rotated through more than 2π along the z axis.

Using CSM we can compute the pose of the cube at its initial position from its image information, and thus determine its initial angular velocity ω_0 and rotation angle θ_0. Also, after each step of motion of the cube, its pose can be determined by the updated image information of the cube. The actual pose of the cube after each manipulation step can be compared with the desired pose of the cube. Since the cube is put on one platform and commanded to rotate along z direction with that platform for an angle of 0.105 radian every step, ω will be $\{0\ 0\ -1\}^T$ for every step when we refer to the pose of the cube. Table 1 shows data for the actual pose, desired pose and error of motion (in radian) after each step in the motion.

We can conclude that the error for each manipulation step is very trivial, the maximum error in the rotation angle amounts to approximately 1 degree. The motion planning method works well with the stratified manipulation of the non-smooth cube object. Although we just show the motion of the cube along z direction, the motion along other direction is also achievable. Just by changing the non-smooth surface parameterization of the object, the method can be generally applied to the stratified manipulation of many non-smooth objects.

Pose	actual θ	desired θ	error
1	0.3178	0.3178	0.0
2	0.2144	0.2128	0.0016
3	0.1034	0.1094	0.0060
4	6.2788	6.2816	0.0028
5	6.1199	6.1738	0.0539
6	5.9946	6.0149	0.0203
⋮	⋮	⋮	⋮
56	0.7542	0.7446	0.0098
57	0.6478	0.6537	0.0059
58	0.5326	0.5428	0.0102
59	0.4313	0.4276	0.0037
60	0.3201	0.3263	0.0062
61	0.2011	0.2151	0.0140
62	0.1022	0.1061	0.0039
63	6.2688	6.2804	0.0115
64	6.1878	6.1638	0.0240
65	6.0993	6.0828	0.0165
66	5.9875	5.9943	0.0068

TABLE 1. Manipulation data.

Furthermore, the experiment with CSM achieves more robustness than simple open-loop control. For the open-loop finger gaiting experiment (see [1]), it is difficult to achieve a long motion along any direction due to the large accumulation of the error from each manipulation step. But for the experiment with vision, we can manipulate the cube for a long motion by updating the pose of the cube after every manipulation step.

7 Conclusions

This paper is an extension of the authors' previous work in stratified motion planning in which a robust, vision-based control strategy is employed to greatly enhance the precision and robustness of the method. An outline of camera-space manipulation, stratified motion planning and the incorporation of the former into the latter is presented. Some experimental results are also presented. Future work includes further demonstration of the method on the more challenging and highly nonlinear case when the shape of the object is more complex than a cube.

Acknowledgements

The support of the National Science Foundation is gratefully acknowledged (Grants No. ISS99-84107-CAREER and IIS99-10602-SGER).

References

[1] Yejun Wei and Bill Goodwine. Stratified motion planning on non-smooth domains with application to robotic legged locomotion and manipulation. In *In Proceedings of the 2002 International Conference on Robotics and Automation*, 2002.

[2] Bill Goodwine and Joel Burdick. Controllability of kinematic systems on stratified configuration spaces. *IEEE Transactions on Automatic Control*, 46(3):358–368, 2001.

[3] Bill Goodwine and Joel Burdick. Trajectory generation for kinematic legged robotics. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. IEEE, 1997.

[4] Bill Goodwine and Joel Burdick. Gait controllability for legged robots. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. IEEE, 1997.

[5] John William Goodwine. *Control of Stratified Systems with Robotic Applications*. PhD thesis, California Institute of Technology, 1998.

[6] Bill Goodwine and Joel Burdick. Stratified motion planning with application to robotic finger gaiting. To appear in the Proceedings of the 1999 IFAC World Congress, 1999.

[7] Bill Goodwine. Stratified motion planning with application to robotic finger gaiting. Proceedings of the 10th World Congress on the Theory of Machines and Mechanisms, 1999.

[8] Bill Goodwine and Joel Burdick. A general method for motion planning for quasi-static legged robotic locomaotion and finger gaiting. In *IEEE Transactions on Robotics and Automation*, 2000.

[9] Bill Goodwine and Yejun Wei. Theoretical and experimental investigation of stratified robotic finger gaiting. In *In Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, 2001.

[10] Yejun Wei, Steven B.Skaar, and Bill Goodwine. Vision-based stratified robotic manipulation. In *IROS,2002*. IEEE, 2002.

[11] E. Rimon and J.W. Burdick. Configuration space analysis of bodies in contact – i. *Mechanism and Machine Theory*, 30(6):897–912, August 1995.

[12] E. Rimon and J.W. Burdick. Configuration space analysis of bodies in contact – ii. *Mechanism and Machine Theory*, 30(6):913–928, August 1995.

[13] K.B. Shimoga. Robot grasp synthesis algorithms: A survey. *The International Journal of Robotics Research*, 15(3):230–266, 1996.

[14] David J. Montana. The kinematics of contact and grasp. *The International journal of Robotics Research*, 7(3):17–32, 1988.

[15] J.C. Trinkle and R.P. Paul. Planning for dexterous manipulation with sliding contacts. *The International Journal of Robotics Research*, 9(3):24–48, 1990.

[16] L. Han, Y.S. Guan, Z.X. Li, Q. Shi, and J.C. Trinkle. Dextrous manipulation with rolling contacts. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 992–997. IEEE, 1997.

[17] Ronald S. Fearing and T.O. Binford. Using a cylindrical tactile sensor for determining curvature. *IEEE Transactions on Robotics and Automation*, 7(6):806–817, 1991.

[18] Kenneth Salisbury and Christopher Tarr. Haptic rendering of surfaces defined by implicit functions. In *Proceedings of the 1997 ASME International Mechanical Engineering Congress and Exposition*, pages 61–67, 1997.

[19] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: Programming touch interation with virtual objects. In *Proceedings of the Symposium on Interactive 3D Graphics*, pages 123–130, 1995.

[20] T. Okada. Object handling system for manual industry. *IEEE Transactions on Systems, Man and Cybernetics*, 9(2):79–89, 1979.

[21] Maw Kae Hor. *Control and Task Planning of the Four Finger Manipulator*. PhD thesis, NYU, 1987.

[22] R.S. Fearing. Implementing a force strategy for object reorientation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 96–102, 1986.

[23] Bill Goodwine and Joel Burdick. Stratified motion planning with application to robotic finger gaiting. In *Proceedings of the 1999 IFAC World Congress*, 1999.

[24] J. Hong and G. Lafferriere. Fine manipulation with multifinger hands. In *IEEE International Conference on Robotics and Automation*, pages 1568–1573, Cincinnati, OH, May 1990.

[25] I-M Chen and J.W. Burdick. A qualitative test for n-finger force-closure grasps on planar objects with applications to manipulation and finger gaits. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 814–820, 1993.

[26] Domenico Prattichizzo and Antonio Bicchi. Dynamic analysis of mobility and graspability of general manipulation systems. *IEEE Transactions on Robotics and Automation*, 14(2):241–258, April 1998.

[27] I.E. Sutherland. Three-dimensional data input by tablet. In *Proc. IEEE*, volume 62, pages 453–461, Apt. 1974.

[28] R. Tsai and R. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. In *IEEE Transaction on Robotics and Automation*, volume 5, pages 345–358, 1989.

[29] R. Tsai. A versatile camera calibration technique for high accuracy 3-d machine vision metrology using off-the-shelf tv cameras and lenses. In *IEEE Transaction on Robotics and Automation*, volume 3, pages 323–344, Aug. 1987.

[30] J. Hill and W.T. Park. Real time control of a robot with a mobile camera. In *Proceeding of 9th ISIR*, pages 233–246, Mar. 1979.

[31] Zachary Dodds, martin Jagersand, Greg hager, and Kentaro Toyama. A hierarchical vision architecture for robotic manipulation tasks. In *ICVS*, pages 312–330, 1999.

[32] S.B. Skaar, W.H. Brockman, and R Hanson. Camera space manipulation. In *The International Journal of Robotics Research*, volume 6(4), pages 20–32, June 1987.

[33] Steven B.Skaar, Issac Yalda-Mooshabad, and William H.Brockman. Nonholonomic camera-space manipulation. In *Proceedings of IEEE Transactions on Robotics and Automation*, volume 8(4), pages 464–478, August 1992.

[34] Steven B.Skaar. Camera-space manipulation. Electornically in `http://www.nd.edu/NDInfo/Research/sskaar/Home.html`, Nov 1994.

[35] H. Morton, J. L. Junkins, and J. Blanton. Analytical solutions for euler parameters. In *Celestial Mechanics*, volume 10, 1974.

[36] W.Z. Chen, U.A. Korde, and S.B. Skaar. Position control experiments using vision. In *The International Journal of Robotics Research*, pages 199–208. MIT Press, June 1994.

[37] Emilio J.Gonzalez-Galvan, Steven B.Skaar, and Michael J.Seelinger. Efficient camera-space target disposition in a matrix of moments structure using camera-space manipulation. In *The International Journal of Robotics Research*, pages 809–818. Sage Science Press, August 1999.