Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Vision-Based Stratified Robotic Manipulation

Yejun Wei ywei@nd.edu
Steven B.Skaar Steven.B.Skaar.1@nd.edu
Bill Goodwine goodwine@controls.ame.nd.edu
Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Abstract— **This paper addresses a three-dimensional implementation that results in several separate manipulators, each with intermittent contact with a central object of known geometry, cooperatively manipulating the object to a desired new position and orientation. Rolling control of an object using redundant contact of several independent manipulators is made difficult by the hybrid nature of this system that introduces complexity to the trajectory-planning problem, and by imperfection in the kinematic models which are needed to achieve such trajectory planning. The former is a theoretical problem which has been solved using Lie-algebra-based strategies to plan motion for the stratified systems. Imperfection in the kinematic models, on the other hand, leads to a practical implementation problem since even small errors in the equations that relate the internal pose of the robot to the position of the end-effector make precise sustained contact with an object difficult. The consequent lack of control of contact force combined with frictional unpredictability associated with rolling, causes gradual growth in the disparity between actual and calculated position and orientation of the object. A robust means for applying vision to compensate for imperfections in the holonomic kinematics of the robots as well as to update estimates of the pose of the object is outlined. Experimental results are also presented.**

1 Introduction

A hybrid system is characterized by two distinct types of interacting subsystems: subsystems with continuous dynamics and subsystems with discrete dynamics. Well-known robotic examples of hybrid systems include legged locomotion and manipulation via finger gaiting. Due to the hybrid nature of such systems, normal control methods applicable to smooth systems can not be directly applied. A general method for motion planning for so-called "stratified systems" is presented in [1],[2],[3] for finger gaiting systems. Experimental results were presented by [4],[5] to apply the method to a real finger gaiting problem.

However, this previous experimental work by the authors is based upon open loop control. After the fingers rotate the object for a given amount, they clearly will need to switch their positions before continuing with the finger gaiting motion. There is no position and orientation information about the object feed-back to the system. Thus, once the manipulation starts, the error from each portion of the manipulation naturally accumulates as the manipulation continues, which obviously limits the experimental verification of the general method for the stratified systems. In this paper, camera-space manipulation (outlined subsequently) is integrated into the manipulation method to provide position and orientation information about the object during the motion. Whenever switching fingers, the pose of the object is updated from the computer vision information obtained about the object's position and orientation.

Robotic grasping and manipulation have been the subject of many research efforts, and only an overview can be provided here. Vast efforts have been directed toward the *analysis* of grasp stability and force closure [6], [7], [8], motion planning assuming continuous contact [9], [10], [11] and haptic interfaces and other sensing [12], [13], [14]. *Finger gaiting*, where fingers make and break contact with the object has been less extensively considered and is the main focus of this paper. Finger gaiting has been implemented in certain instances [15], [16], [17] and also partially considered theoretically [18], [19], [20].

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, problems with three-dimensional implementations of such a strategy have centered on difficulties with achieving and sustaining accuracy of the calibration of both the robots and cameras [21],[22]. This problem becomes particularly acute when high levels of precision are required due to the need to preserve contact without exerting an undue contact force between the fingers and object.

A second approach to vision-based robotic control is visual servoing [23],[24]. Here, errors between the current pose of the robot end effector and the desired finger contact location on the surface of the object are computed directly in the cameras. The Jacobin which relates incremental movement of a given robot's joint coordinates to the image-plane response is utilized to "servo" all image-plane errors toward zero. While ro-

0-7803-7398-7/02/$17.00 ©2002 IEEE

bust to even relatively large errors in the Jacobin, this strategy is problematic for the following reason. Due to the nonholonomic nature of the relationship between the motion of each robot and the response of the manipulated object, the terminal pose of the manipulated body depends upon the trajectory followed by each of the robots; therefore, the terminal attitude of the object is not merely a function of the robots' terminal poses. So any intermediate visual-servoing correction must actually be factored into an updated trajectory plan to produce the desired orientation outcome. On a more practical note, visual servoing requires visual access to contact errors which is particularly difficult when extremely close proximity of the bodies is required over the duration of the trajectory. control requires specified end-member the otherwise undeformed surface location is a very real question as to whether a detected at all.

Camera-space manipulation (CSM) ([25]) is utilized in order to avoid some of the difficulties of calibration or visual servoing. As with visual servoing, CSM treats the problem as one of achieving maneuver success in the image planes of the cameras. However, as outlined subsequently, it does so with trajectory planning of the robots' joint rotations based not upon *a priori* physical kinematics of the robot but rather real-time-estimated and refined "camera-space kinematics." The result is the ability to sustain contact between each robot and the object throughout the duration of a complex maneuver. Of course the trajectory planner must update the trajectory periodically due to actual object slippage and kinematics imperfection, but such updates are less frequent due to the system's ability to compensate for imperfections in the robot's nominal physical kinematics model. Moreover, the concept of CSM can be integrated into the Lie-algebraic based trajectory planner without significant modification.

2 Stratified Manipulation

This work is an extension of previous work by the authors; therefore, a short review of previous results is necessary. Many details are necessarily omitted, and the interested reader is referred to [1], [2], [3], [4], [5], [26], [27], [28], [29] for a complete, detailed exposition.

A simple example will provide an intuitive understanding of the geometry inherent in stratified systems. Consider the simplistic example of two fingers intermittently engaging a smooth object, such as a sphere. The set of configurations corresponding to one of the robots engaging the object is a smooth codimension one submanifold contained in the configuration space. The same is true when the other robot engages the object. Similarly, when both robots engage the object, the system is on a smooth codimension two submanifold of the configuration space formed by the intersection of the single contact submanifolds. Each submanifold is referred to as a *stratum*. The structure of the configuration manifold for such a system is abstractly illustrated in Figure 1.

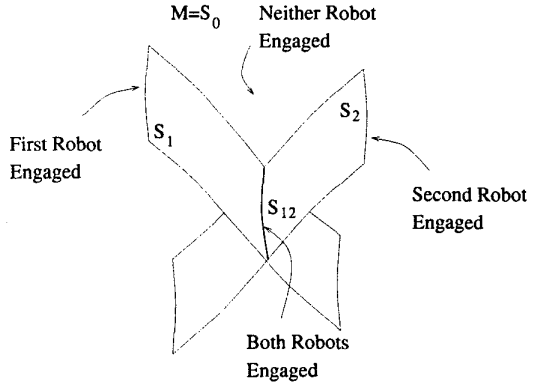

Figure 1. Configuration manifold structure for two cooperating robots.

More generally, if there are n robots engaging a common object, let S_0 denote the system's entire configuration manifold and $S_i \subset S_0$ the codimension one submanifold of S_0 that corresponds to all configurations where only the ith robot engages the object. Denote $S_{ij} = S_i \cap S_j$. The set S_{ij} physically corresponds to states where both the ith and jth robots engage the object. Further intersections can be similarly defined in a recursive fashion: $S_{ijk} = S_i \cap S_j \cap S_k = S_i \cap S_{jk}$, etc. The lowest-dimensional stratum (corresponding to all fingers in contact with the object) will be called the *bottom stratum.*

We assume that the equations of motion on each stratum, S_I, are of the form

$$\dot{x} = g_{I,1}(x)u_{I,1} + \cdots + g_{I,n_I}(x)u_{I,n_I}, \qquad (1)$$

where the first subscript, I, indexes the stratum upon which the equations are defined. And the second subscript, n_I, dependes upon the codimension of S_I and the nature of the additional constraints imposed on the system in S_I.

The motion planning algorithm for smooth stratified systems is based upon the method presented in [30]. The approach is to construct an *extended system* in which the original set of equations of motion is appended with Lie bracket vector fields associated with which are *fictitious* inputs. For the extended system, motion planning is trivial since it is constructed so that the span of all the vector field is full rank. Formal algebraic computations utilizing indeterminates, b_i, formal exponential expansions of the

form $e^{b_i} = 1 + b_i + \frac{b_i^2}{2!} + \cdots$, which can be related to solutions of the original equations (1) and approximations to Lie brackets of the form

$$\phi_\epsilon^{-g_2} \circ \phi_\epsilon^{-g_1} \circ \phi_\epsilon^{g_2} \circ \phi_\epsilon^{g_1}(x) = \phi_{\epsilon^2}^{[g_1,g_2]}(x) + \mathcal{O}(\epsilon^3), \quad (2)$$

where $\phi_\epsilon^{g_1}(x_0)$ represents the solution of the differential equation $\dot{x} = g_1(x)$ at time ϵ starting from x_0 provide a mechanism to determine the real control inputs.

For stratified system, if it is the case that the Lie bracket between the vector fields which switch the system among strata and any other vector fields is zero (called "Lie bracked decoupling"), then it is straight forward to show that vector fields defined on *multiple strata* can be considered simultaneously in the motion planning algorithm (a detailed explanation can be found in [29]). An outline of the algorithm is as follows.

1. Check that the Lie bracket decoupling assumption holds.
2. Check that the stratified system is controllable (see [2], [26]).
3. Determine a *nominal trajectory* in the bottom stratum.
4. Construct the *extended stratified system* on the bottom strata. This is of the form

$$\begin{aligned} \dot{x} &= g_1(x)v_1 + \cdots g_m(x)v_m \\ &+ \underbrace{g_{m+1}v_{m+1} \cdots + g_n v_n}_{\text{from higher strata}} \\ &+ \underbrace{g_{n+1}v_{n+1} + \cdots + g_p v_p}_{\text{any Lie brackets}}, \end{aligned} \quad (3)$$

   where the $\{g_1, \ldots, g_p\}$ span $T_x S_0$ and are the control vector fields from multiple strata, the inputs $v_1, \ldots, v_n$ are real, and the inputs $v_{n+1}, \ldots, v_p$ are *fictitious.*
5. Construct the *formal equation*, which is simply Equation 3 written in indeterminates, $\dot{S}(t) = S(t)(b_1 v_1 + \cdots + b_s v_s)$, where the $S(t)$ are polynomial Lie series (see [30] for details).
6. Construct the Chen-Fleiss series, namely, $S(t) = e^{h_s(t)b_s} e^{h_{s-1}(t)b_{s-1}} \ldots e^{h_1(t)b_1}$, differentiate it with respect to time and equate the coefficients of the b_i's in the resulting equation with the coefficients of the corresponding b_i's in the equation in the previous step, to construct ordinary differential equations for the *backward Philip Hall coordinates*, h_i.
7. Solve the o.d.e.'s from the previous step to determine the h_i's to determine how long the system should flow along each basis element, b_i, to reach the goal point. If the b_i represents a Lie bracket, then an approximation of the form of Equation (2) should be used.
8. If two sequential b_i's belong to different strata, then the decoupled vector field (checked in step 1) must be actuated to switch strata.

Unfortunately, most of these steps are rather involved, but space limitations prevent the inclusion of most detail. Again, references [30], [31] provide a good overview of the smooth version of the algorithm, and references [1], [2], [3], [4], [26], [27], [28], [29] present the extension to stratified systems.

3 Camera-Space Manipulation

The camera-space manipulation method 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 contains a lens that can form a projection from the 3D Cartesian (physical) space in view onto the cameras' 2D image plane as shown in Figure 2.

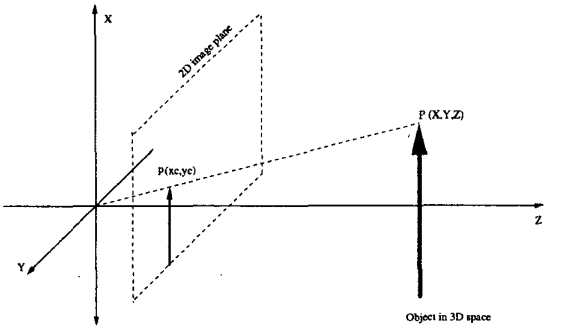

Figure 2. Mapping from the Cartesian coordinates to the image plane coordinates

The camera model used here is a perspective projection model. As shown in Figure 2, 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 ([25],[34]). 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

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = F(X, Y, Z; \vec{C}), \tag{4}$$

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 ([35],[32]), 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:

$$\begin{aligned} J(\vec{C}) = \sum_{i=1}^{n_p} \Big[\sum_{j=1}^{n_{ci}} \Big\{ &(x_{c_{ij}} - F_x(X_i, Y_i, Z_i; \vec{C}))^2 \\ &+ (y_{c_{ij}} - F_y(X_i, Y_i, Z_i; \vec{C}))^2 \Big\} W_j \Big] W_i, \end{aligned} \tag{5}$$

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 cues detected in one camera sample. This algorithm is based on the orthographic camera model, which requires that all the interested physical cues are close to each other. A flattening process is introduced in [25], which gets rid of this constraint, and give 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 (4). 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:

$$\begin{bmatrix} x_{c_{i_1}} \\ y_{c_{i_1}} \\ \vdots \\ x_{c_{i_m}} \\ y_{c_{i_m}} \end{bmatrix} = G(X_i, Y_i, Z_i, \vec{C}_1, \cdots, \vec{C}_m), \tag{6}$$

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 Stratified Manipulation Algorithm

The flow chart illustrated in Figure 3 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 5. The nonunique 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 6.

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.

5 Experimental Implementation

An experiment has been carried out to show the application of Camera-Space Manipulation method to the stratified finger gaiting experiment based on the algorithm in Figure 3. Four Puma 560 robots are mounted on one common platform, where each 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 II 500 MHz PC running Linux operating system to control the mo-

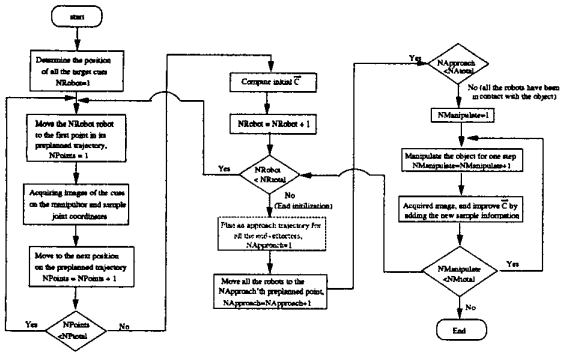

Figure 3. Flow chart representing the experiment procedure.

tion of the 24 joints of the 4 robots. A picture of the experimental platform is illustrated in Figure 4.

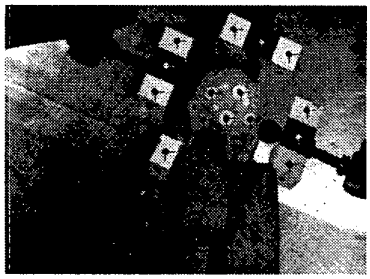

Figure 4. Experiment Setup

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.

Although the method is general in that it can accommodate any smooth or polygonal object [5], in this paper we will primarily consider a spherical object. as the common object for the four fingers to cooperatively manipulate. A unique black cue and 38 white cues, with each cue at a known position relative to the ball, are attached to the ball. Once the view parameters are initialized, the cue images in the image plane can be distinguished and matched to the cues on the ball object and the manipulator, and the position and orientation of the ball 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 ball about an axis oriented in the $x-z$ direction with an angular velocity $\{0.287, 0.0, -0.958\}^T$ 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 ball. After each step of motion, samples are again acquired, and the $\vec{C}$'s are correspondingly updated. The result for the above motion is shown in the next section. Also, the motion about axes oriented in other directions like z, $y-z$, $x-y-z$ are also carried out, and obtainable as well.

6 Experimental Results

The pose of the ball as well as the manipulators during motion are shown in the Figures 5 through 10. Using CSM method, we can compute the pose of the ball during the manipulation from the image information of the ball, and thus determine the angular velocity ω and rotation angle θ of that pose relative to the robot frame as follows:

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

Hence,

$$\theta = \cos^{-1}\left(\frac{t_{11}+t_{22}+t_{33}-1}{2}\right),$$

$$\omega = \frac{1}{2\sin(\theta)}\begin{bmatrix} t_{32}-t_{23} \\ t_{13}-t_{31} \\ t_{21}-t_{12} \end{bmatrix}.$$

Also, if the object experiences a rotation of θ along axis ω from its initial pose T_i, its final pose would be:

$$T_f = e^{\hat{\omega}\theta} T_i.$$

Thus, we can compare the actual pose of the ball with the desired pose during the manipulation. A pose of a $3D$ object consists of 6 variables: 3 orientation variables and 3 position variables. Figure 12 presents the comparison of the actual and desired orientation of the

Figure 5. $\theta = 0.0$ radian

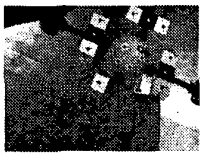

Figure 6. θ = 1.15 radian

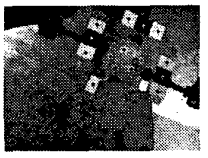

Figure 7. θ = 2.30 radian

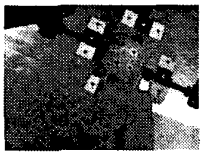

Figure 8. θ = 3.45 radian

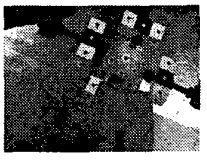

Figure 9. θ = 4.60 radian

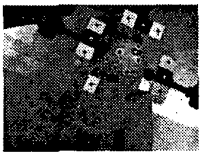

Figure 10. θ = 5.75 radian

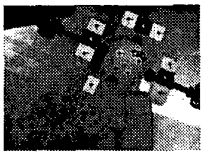

Figure 11. $\theta = 6.80$ radian

object, and Figure 13 presents the comparison of the position of the object during the motion.

We can conclude that there is some error in the motion. It is acceptable since the ball object is compressible, and there is some unpredicted translation and rotation of the object when switching fingers, and the error accumulates all through the finger gaiting process, where there are $63 \times 4 = 252$ times finger switching.

Furthermore, the experiment with CSM achieves more robustness than simple open-loop control. For the open-loop finger gaiting experiment (see [5]), it is difficult to achieve a long motion along pure z direction due to the reason that the friction forces from the contact between the four finger with the ball object is not always enough to hold the ball. But for the experiment with vision, we can manipulate the ball in most directions specified.

7 Conclusions

This paper is an extension of the authors' previous work in stratified motion planning in which a robust,

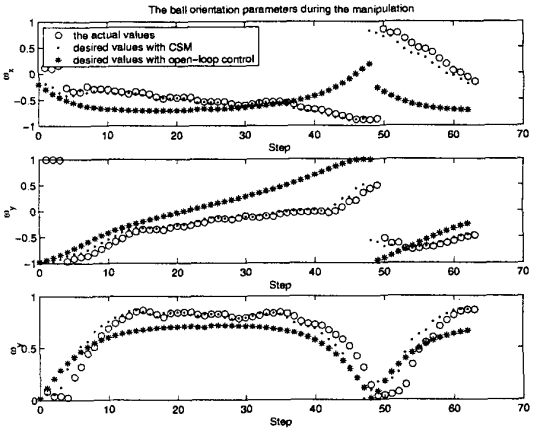

Figure 12. Comparison of the orientation of the object during motion.

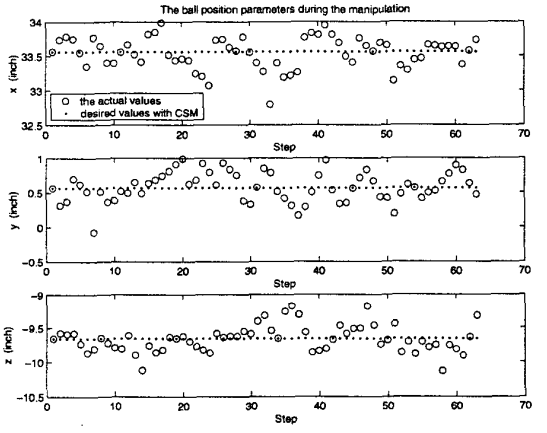

Figure 13. Comparison of the position of the object during motion.

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 objects are polygonal.

Acknowledgements

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

References

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

[2] Bill Goodwine and Joel Burdick. Controllability of kine-

matic 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 robots. In *IEEE International Conference on Robotics and Automation*, pages 2689–2696, Albuquerque, NM, 1997.

[4] 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.

[5] 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.

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

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

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

[9] David J.Montana. The kinematics of contact and grasp. In *The International Journal of Robotics Research*, volume 7, pages 17–32, 1988.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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

[17] 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.

[18] 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.

[19] 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.

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

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

[22] 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.

[23] 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.

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

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

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

[27] 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.

[28] 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.

[29] 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.

[30] G. Lafferriere and H.J. Sussmann. Motion planning for controllable systems without drift. In *In Proceedings of the IEEE Conference on Robotics and Automation*, pages 1148–1153. IEEE, 1991.

[31] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

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

[35] 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.