Uncalibrated Hybrid Force-Vision Manipulation

Andreas Pichler, Martin Jägersand
CS Dept, Johns Hopkins University, Baltimore, MD 21218
{pichler,jagersand}@cs.jhu.edu

Abstract

We present a method employing hybrid force and vision based control to effect a sequence of contact manipulations. Instead of requiring a-priori object and environment models, force sensing is used to simulataneously update a surface model while controlling the manipulator. This is incorporated into an uncalibrated visual servoing system, which also estimates the visual-motor coordinate transform. The result is a hybrid force-vision controller which does not need any a-priori robot, camera, object or environment models. The approach is validated experimentally using an IMI Zebra robot arm.

1 Introduction

Robots have successfully been employed in industrial automation for several decades, and a variety of both manipulators and control methods have been developed. Despite much interest, however, success has been limited outside of engineered industrial or laboratory environments. A pitfall has been the need for detailed and accurate a-priori models of both the robots and the environment. In most everyday environments (e.g. in normal homes, outdoors and in most work places designed for people) exact modeling is typically too tedious to be practical, and undesirable because it would impose a rigorous regiment on human behavior to not move things around or act in a way that violates the model.

Computer vision can provide a powerful way of sensing the environment and can potentially reduce or obliterate the need for environmental modeling. Research in visual feedback control has been very active in the last decade; first for allowing visual end point feedback (e.g. [2, 3, 6]) and later to also involve estimating the needed visual-motor models on-line (e.g. [4, 5]). Visual servoing has proven to be a way of performing accurate movement in free space without the need for accurate or even any a-priori models [12]. It has been more difficult to use to solve whole manipulation tasks. Without an accurate global world model, task planning and decomposition is instead better done in visual space [11].

While uncalibrated visual servoing is suitable for free space movements, many tasks are more easily solved though a sequence of contact manipulations. Examples include most insertion or assembly tasks. When inserting things humans typically rely on both visual and tactile feedback. Removing either reduces performance. In robotics the problem of simultaneous "hybrid" force-position control has been studied in calibrated settings [8, 13]. However, this work does not directly transfer to uncalibrated settings because the constraint geometry is unknown, and has to be learned. In addition, if the manipulator is also uncalibrated, methods have to be reformulated in hybrid sensory spaces instead of relying on a common Cartesian robot base frame. Some work has been started in the direction of combining visual servoing with force control, but approaches so far require the task space to be pre-partitioned into a force and visual part[7, 9].

The main contribution of this paper is to show how the constraint geometry can be learned and refined during contact manipulations, and how transforms between the two sensory (force and vision), motor and constraint frames can be learned on-line. Other sensory based approaches can be found in [14, 15]. The main difference is that we focus on a formulation to facilitate switching between the number and types of constraints. The particular tasks we are interested in are ones in which an increasing number of force constraints are imposed during motion. To intuitively illustrate the usefulness of this, consider Fig. 1 in which a corner to corner alignment is performed with a cube. In a free-space approach the two corners would normally be aligned using one direct movement, and the final positioning would normally be no more accurate than if there had been no physical reference. A more robust strategy is to take advantage of the surfaces during the manipulation, and define a sequence which successively impose more surface constraints. In the contact sequence the first movement need only reach some place on the surface, then motion is constrained to along the surface and a sliding contact motion is used to translate to the corner. When reaching the corner similarly contact is maintained with both sides while a rotating motion is used to align the corners. Note that each one of these motions need little positional accuracy, hence the sequence of contact manipulations is more robust than the free-space positioning approach. Many insertion and assembly motions can be decomposed in a similar way to a sequence of robust contact manipulations.

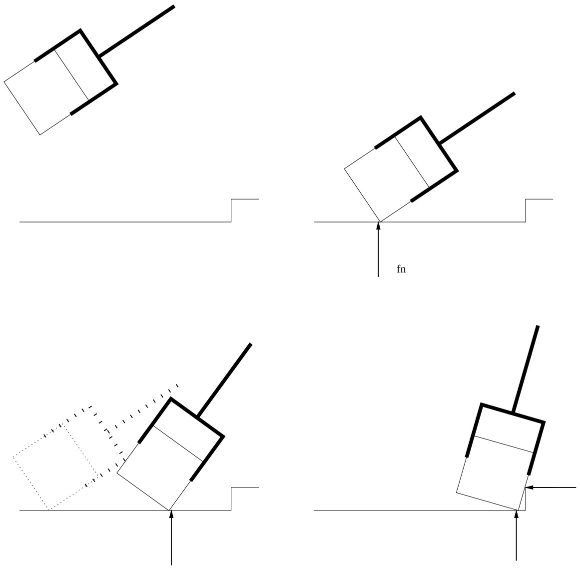

Figure 1: Aligning a cube with a corner.

The rest of the paper is structured as follows. In the next section we start by reviewing uncalibrated visual servoing and how to acquire the visual-motor transforms, then we define the needed surface geometry needed for hybrid force-vision control, and show how it can be learned using contacts and movements in the force sensor frame. Finally we extend the controller to a hybrid force vision controller where the freedoms are split into vision and force control along the directions defined by the constraint surface. In the experiments section we show how to perform contact manipulations and transitions between different numbers of constraints using an IMI Zebra manipulator.

2 Theory

Visual information originates in one or more (e.g. stereo) camera frames. Contact force is most often measured by a force-torque sensor on the end effector, and hence is given in a moving tool-frame. Hybrid visual-force manipulation depends on relating three coordinate frames: the visual frame, the force sensor frame and the constraint frame defined by the contact surface.

2.1 Preliminaries: Visual Servoing

We study a robot hand-eye system, in an unstructured environment. The robot action reference frame is joint space, described as desired joint angles, $\mathbf{x} = (x_1 \dots x_n)^T$, and their time derivatives $\dot{\mathbf{x}}, \ddot{\mathbf{x}}$. The sensory input is intensity images $\mathbf{I}$. We don't use these directly. Instead a much reduced space of *feature* vectors $\mathbf{y} = (y_1 \dots y_m)^T$ are tracked to describe the visual state during manipulation, where the components of $\mathbf{y}$ are the row and column image projections of points on a manipulated object. The composition of camera calibration, robot kinematics, object shape and their relative geometry defines a non-linear visual motor function f, relating visual and motor frames. In an uncalibrated setting f is not known, and instead we approximate f linearly: $f(\mathbf{x}) \approx f(\mathbf{x}_k) + J(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)$. The model is valid around the current system configuration $\mathbf{x}_k$, and since $f(\mathbf{x}_k) = \mathbf{y}_k$ is the current tracker state the only unknown is the "image"[6] or visual-motor Jacobian J. We have previously shown how the Jacobian can be estimated form observation only using for instance a rank 1 correction term:

$$\hat{J}_{k+1} = \hat{J}_k + \frac{(\Delta \mathbf{y}_{measured} - \hat{J}_k \Delta \mathbf{x})\Delta \mathbf{x}^T}{\Delta \mathbf{x}^T \Delta \mathbf{x}} \tag{1}$$

From this it is straightforward to derive a visual motion controller, e.g. a P-controller of the form:[1]

$$\dot{\mathbf{x}} = K J_k^+ \dot{\mathbf{y}} \tag{2}$$

Where J^+ is a pseudoinverse of J. In [12] we evaluate such a controller through quantitative accuracy experiments, and in [11] we show how to use it to solve real world manipulation tasks by planning all movements in free space. In what follows we will extend the uncalibrated approach to also allow contact manipulation.

2.2 Acquiring Constraint Geometry

In hybrid force-vision control we need to split the control space so that vision based position and velocity control is limited to dimensions along non-contact directions, while force control is used to maintain desired contacts. To do this in an uncalibrated framework we have to find the constraint surface, and its relation to the visual (camera coordinates) and force (tool) frames.

Let P be a basis for the differential surface geometry of a contact point, with basis vectors $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ in the tool frame $\mathcal{T}$. We explore the surface using the force sensor information. Here we assume the force sensor is mounted on the end-effector, so the force space is the same as the robot tool frame.

From physics we know that an ideal impulse contact results in a reaction force normal to the contact surface. In practice impacts with most normal objects are close enough so that at the transition from free space manipulation to a point impact at time k we can read force $\mathbf{f}_k \in \mathcal{T}$ and define

$$\mathbf{p}_1 = \frac{\mathbf{f}_k}{|\mathbf{f}_k|}$$

in tool-frame coordinates (The force sensor is affixed to the end effector). Note that for now we consider a frictionless environment and zero gravity.

[1] Control and model acqusition do not have to run in the same loop. In fact the model updating is more accurate if performed only after several pixels image movement, while control should be run at as close to frame rate as possible.

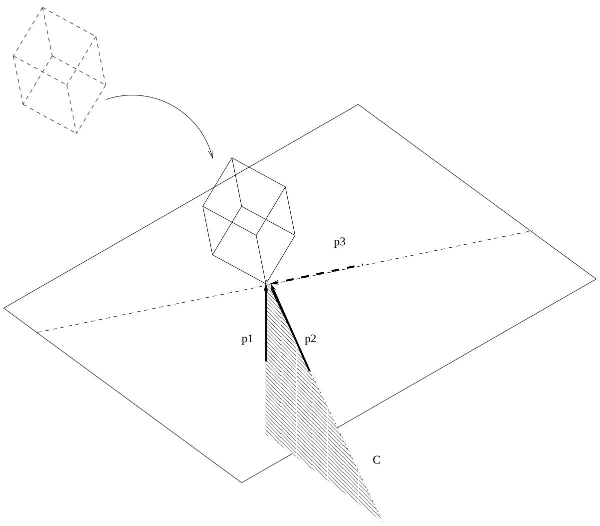

Figure 2: Directions of constraint vectors.

The second constraint vector is defined to align with the differential movement $\delta\mathbf{x} \in \mathcal{T}$ along the surface:

$$\mathbf{p}_2 = \frac{\delta\mathbf{x}_k}{|\delta\mathbf{x}_k|}$$

Both $\mathbf{p}_1$ and $\mathbf{p}_2$ are orthonormal vectors so we can get the third constraint vector $\mathbf{p}_3$ in a three dimensional Cartesian space by:

$$\mathbf{p}_3 = \mathbf{p}_1 \times \mathbf{p}_2 = \frac{\mathbf{f}_k}{|\mathbf{f}_k|} \times \frac{\delta\mathbf{x}_k}{|\delta\mathbf{x}_k|}$$

Each of these measurements is available at any sample time k, hence we define an updating rule for the mapping from tool to constraint space $P : \mathcal{T} \rightarrow \mathcal{C}$ as follows:

$$P_{k+1} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \begin{pmatrix} \frac{\mathbf{f}_k}{|\mathbf{f}_k|} \\ \frac{\delta\mathbf{x}_k}{|\delta\mathbf{x}_k|} \\ \frac{\mathbf{f}_k}{|\mathbf{f}_k|} \times \frac{\delta\mathbf{x}_k}{|\delta\mathbf{x}_k|} \end{pmatrix}$$

Pdefines a constraint space denoted by $\mathcal{C}$ and we hereforth write vectors in $\mathcal{C}$ with prime, e.g. x'.

2.3 Hybrid Control Law

Consider again a purely visual controller of the form in Eq. 2. During contact manipulation we split the control space according to the constraint space. Let Q be a joint space to tool-frame Jacobian, $Q : \mathcal{J} \rightarrow \mathcal{T}$, and with P defined as before we constrain the visual control with a selection matrix S. In the constraint space S is a diagonal matrix, e.g. of the form

$$S = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for 3D control with 2 visual and 1 force freedom.

Since the relation between changes of feature space $\dot{\mathbf{y}}_k$ and changes of the motor space $\dot{\mathbf{x}}_k$ is known, we can set up the geometric mappings for a visual control law:

$$\dot{\mathbf{x}}'_{v_c,k} = S\dot{\mathbf{x}}'_{v,k} = SP_kQ^{-1}J^{-1}\dot{\mathbf{y}}_k$$

where $\dot{\mathbf{x}}'_{v_c} \in \mathcal{C}$. It can be mapped back to the joint space $\mathcal{J}$ with

$$\dot{\mathbf{x}}_{v_c,k} = Q^{-1}P_k^{-1}S\dot{\mathbf{x}}'_{v,k} = Q^{-1}P_k^{-1}SP_kQJ^{-1}\dot{\mathbf{y}}_k$$

where $Q : \mathcal{J} \rightarrow \mathcal{T}$.
This allows us to set up a control law for the vision loop:

$$\mathbf{u}_{v_c,k} = -K_vQ^{-1}P_k^{-1}SP_kQJ^{-1}\mathbf{e}_k$$

where $\mathbf{u}_{v_c,k}$ represents the constraint visual velocity command in joint space coordinates. $\mathbf{e}_k$ is the image feature error between the moving object feature and goal in the image space $\mathcal{I}$.

The force part of the constraint space $\mathcal{C}$ is controlled by a force loop. First the constraint vectors $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ have to be computed from the sensory input.
Depending on the task, one or more freedom in C has to be constrained, by the force loop (I = identity matrix):

$$\mathbf{u}'_{c,k} = -K_c(I - S)P_k\mathbf{f}_k$$

where $\mathbf{u}'_{c,k} \in \mathcal{C}$, K_c the gain factor of the force loop and $I - S$ defines the freedoms controlled by the force loop. To move the robot commands have to be transformed to joint space $\mathcal{J}$

$$\mathbf{u}_{c,k} = -K_cQ^{-1}P_k^{-1}(I - S)P_k\mathbf{f}_k$$

Hence the complete hybrid system can be written as:

$$\begin{aligned} \mathbf{u}_k &= \mathbf{u}_{v_c,k} + \mathbf{u}_{c,k} = \\ &= -K_vQ^{-1}P_k^{-1}SP_kQJ^{-1}\mathbf{e}_k - K_cQ^{-1}P_k^{-1}(I - S)P_k\mathbf{f}_k \end{aligned} \tag{3}$$

where $\mathbf{u}_k \in \mathcal{J}$ the moving command in velocity mode. The block diagram of the resulting hybrid controller is shown in Fig. 2.3

2.4 Non-frictionless environment

Until now we have assumed a frictionless environment. Hence, when sliding along a surface we could directly retrieve the normal force. In a non-frictionless environment we read a force $\mathbf{f}_k$ consisting of two components, the normal force $\mathbf{f}_{n,k}$ and the friction force $\mathbf{f}_{f,k}$ which is perpendicular to $\mathbf{f}_{n,k}$.

Since we let the normal force $\mathbf{f}_{n,k}$ provide the first constraint direction $\mathbf{p}_1$ we have to first subtract $\mathbf{f}_{f,k}$ from the force measurement $\mathbf{f}_k$, hence:

$$\mathbf{p}_1 = \frac{\mathbf{f}_k - \mathbf{f}_{f,k}}{|\mathbf{f}_k - \mathbf{f}_{f,k}|}$$

Since we aligned the constraint coordinate system with $\mathbf{p}_2$ along the motion direction:

$$\mathbf{f}_{f,k} = -f_{f,k}\mathbf{p}_2 = \mathbf{p}_2^T \mathbf{f}_k \mathbf{p}_2$$

Finally, we can complete the friction corrected transform P_{k+1} as:

$$P_{k+1} = \begin{pmatrix} \frac{\mathbf{f}_k - \mathbf{p}_2^T \mathbf{f}_k \mathbf{p}_2}{|\mathbf{f}_k - \mathbf{p}_2^T \mathbf{f}_k \mathbf{p}_2|} \\ \frac{\delta \mathbf{x}_k}{|\delta \mathbf{x}_k|} \\ \frac{\mathbf{f}_k - \mathbf{p}_2^T \mathbf{f}_k \mathbf{p}_2}{|\mathbf{f}_k - \mathbf{p}_2^T \mathbf{f}_k \mathbf{p}_2|} \times \frac{\delta \mathbf{x}_k}{|\delta \mathbf{x}_k|} \end{pmatrix} \tag{4}$$

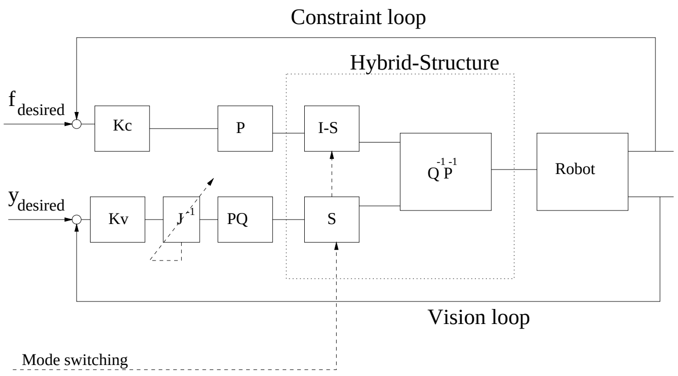

Figure 3: Hybrid control structure

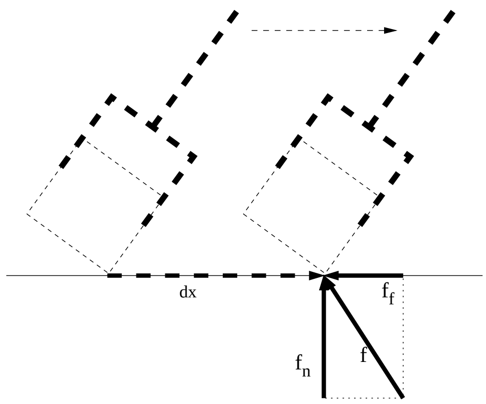

Figure 4: Splitting up force vector

3 Experiments

M-peg video demonstrations of the method can be found in: `http://www.cs.jhu.edu/~ jager/iros/`

We implemented an experimental contact manipulation system on an IMI Zebra robot controlled by a 66MHz 486 linux PC and a 200Mhz Pentium linux PC with Matrox frame-grabbers. Communication, control and tracking programs were written in C using PVM (Parallel Virtual Machine) libraries to communicate between the machines. High level geometric routines and graph generation (for this paper) were written in matlab and interfaced to the PVM program using cmex calls. The control loop was run at approximately 200ms cycle time. The visual processing and the setup with two networked computer were the most limiting factors on the control loop speed.

Figure 5 illustrates the experimental setup and sequence of movement. The Zebra arm holds an arbitrary, unknown object. We tested with both a cylindrical and square shape sorter puzzle piece. The white dots are tracked and image coordinate positions $\mathbf{y}$ are reported to the controller. Image goal configurations are given by the user by clicking with the mouse on displayed camera images.

In the first movement (in free space), motor joint velocities are computed using a pure visual controller, Eq. 2. The transition between free space movement and contact is detected as a step change in the force sensor readings. (Readings are never 0 and vary with robot pose due to gripper and object weight). The first impact force vector is used to set up the constraint frame. Control proceeds with the hybrid control law, Eq. 3, and during contact motion the constraint frame is continuously improved by the updating rule Eq. 4. This will also allow us to handle the curved surface in Fig. 8 When the object reaches the visually defined position on the surface, force control is switched from maintaining just a normal force constraint, to maintaining the achieved point contact by controlling also the forces in the surface plane (to maintain a zero setpoint). The remaining visual freedoms are controlled to rotate the object into alignment with the surface.

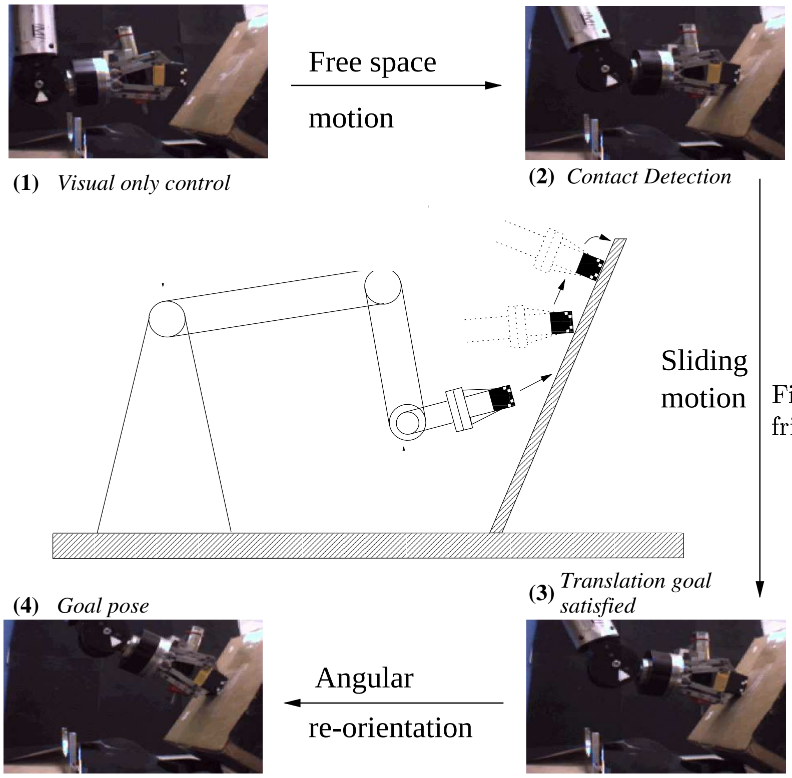

Figure 5: Movement sequence in the experiments. ($1 \rightarrow 2$): Free space control using only the visual controller. ($2 \rightarrow 3$): Visual feedback controls translation.One normal force constriand maintains contact. ($3 \rightarrow 4$) Vision controls rotation. Both both surface and normal force is used to maintain a fixed point contact.

During the sliding part of the motion, the constraint frame is based on the force reading vector and the motion vector. In Fig. 6 a small part of the sliding trajectory is drawn as it was tracked in camera coordinates. Several of the read force vectors are projected onto the image using the same estimated coordinate transforms as in the controller. The subtraction of the friction component and setting up of the constraint frame is illustrated on the middle force vector.

The accuracy of the estimated constraint frame was estimated by comparing the image projection of the estimated surface normals ($\mathbf{p}_1$) to a reference normal vector obtained using the image projection of a stick taped on in the normal direction. The surface normal estimate was typically within a few degrees of the true normal. The measured error distribution is shown in Fig. 7.

Since the constraint frame is updated in every step based on current force and movement information it is

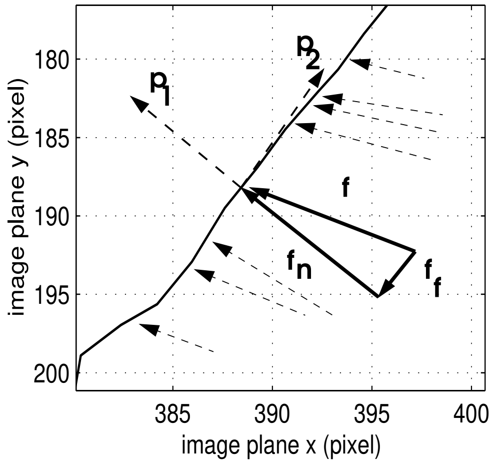

Figure 6: Force reading vectors $\mathbf{f}$ are compensated for friction $\mathbf{f_f}$ to form the constraint space $(\mathbf{p}_1, \mathbf{p}_2, \ldots)$.

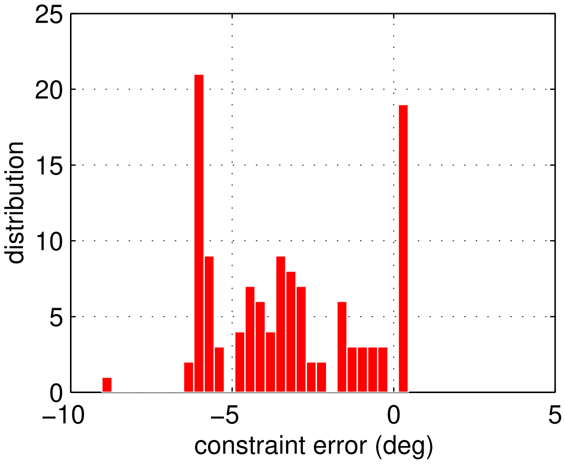

Figure 7: Angle error of the constraint vector $\mathbf{p}_1$ compared to the normal direction of the surface.

also possible to slide along curved surfaces. To experimentally verify this a piece of vinyl floor tile was bent to an curved shape, and the experiment outlined in Fig. 5 was repeated for the curved surface. The resulting trajectory along with the estimated surface normals are displayed in Fig. 8. The few erroneous vectors are results from the manipulator temporarily losing contact with the surface, either due to small bumps on the surface or minor instabilities in the controller.

4 Discussion

We have shown how a constraint geometry for hybrid force-vision manipulation can be set up using only sensory data. This extends current visual servoing capabilities to also being able to perform contact manipulations in unmodeled environments. In the experiments, we showed that we can obtain the constraint surface

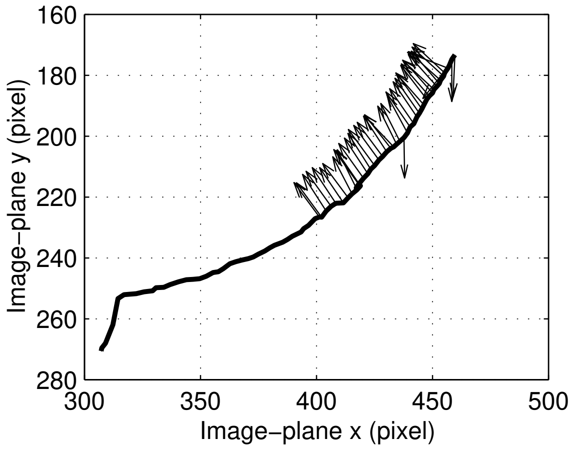

Figure 8: Sliding along curved surface.

with a few degrees accuracy, and showed that this is enough to perform contact manipulations. We implemented a simple controller just for this purpose. A better controller would have to have two loop frequencies; a fast force loop, and a slower vision loop (slower due to camera frame rate and processing limitations). The main points of this paper are the acquisition of the surface geometry, and how that can be naturally integrated into many manipulations by successively increasing the number of surface/force constraints. We refer readers interested in dynamics and other properties of hybrid control to the literature (e.g. [15]) since that is not the main point of this paper.

In the future we plan to apply this method to solving insertion and assembly tasks. In a previous project we have worked with a shape sorting task where 3-5cm big pieces of different shapes had to be matched and inserted into slots about 1mm bigger than the pieces. This turned out to be quite hard to solve using visual servoing alone[11]. (By aligning the piece a few mm above the hole and attempting to insert it by a straight motion down.) Instead a contact manipulation strategy where first one corner of the piece is inserted into the hole with coarse precision, then moved towards an edge, slid towards a corner, rotated to a line to line contact, and finally the whole piece is inserted in a plane to plane aligning movement. In addition to hybrid control this task also requires geometric reasoning to plan a suitable contact sequence, and this is the next issue we plan to address.

References

[1] On-line m-peg movies of the experiments are availible. Use the menu in: http://www.cs.jhu.edu/~jager/iros/

[2] Weiss L. E. Sanderson A. C. Neumann C. P. "Dynamic Sensor-Based Control of Robots with Visual Feedback" *J. of Robotics and Aut.* v. RA-3 1987

[3] Espiau B. Chaumette F. Rives P. "A New Approach to Visual Servoing in Robotics" *IEEE Tr. on Robotics and Automation* p 313-326 v 8 no 3, 1992.

[4] Jägersand M. Nelson R. *Adaptive Differential Visual Feedback for uncalibrated hand-eye coordination and motor control* TR# 579, U. of Rochester 1994.

[5] Hosoda K. Asada M. "Versatile Visual Servoing without Knowledge of True Jacobian" *Proc. IROS* 1994.

[6] Corke P. I. *High-Performance Visual Closed-Loop Robot Control* PhD thesis U of Melbourne 1994.

[7] Castano A. Hutchinson S. "Visual Compliance: Task -Directed Visual Servo Control" In *IEEE Trans. on Robotics and Automation* v.10 nr.3 June 1994, p334-342

[8] Khatib O. "Inertial Properties in Robotic Manipulation: an Object-Level framework" In *Int. Journal of Robotics Research* v. 14 nr.1, Feb 1995.

[9] Nelson B. Khosla P. *Force and Vision Resolvability for Assimilating Disparate Sensory Feedback* CMU-RI-TR95-11, Carnegie Mellon University Feb. 1995

[10] Kutulakos K. Jägersand M. "Exploring objects by purposive viewpoint control and invariant-based hand-eye coordination" *Workshop on vision for robots* In conjunction with IROS 1995.

[11] Jägersand M. Nelson R. "Visual Space Task Specification, Planning and Control" In *Proc. IEEE Int. Symp. on Computer Vision*, 1995.

[12] Jägersand M. Nelson R. Fuentes O. "Experimental Evaluation of Uncalibrated Visual Servoing for Precision Manipulation" In *Proc. of Int. Conf. on Robotics and Automation* 1997.

[13] Schutter J, Bruyninckx H, Spong M. "Force control: a birds eye view" In *IEEE Int. Workshop on Control Problems in Robotics and Aut"* 1997

[14] Hosoda K. Igarashi K. Asada M. "Adaptive Hybrid Control for Visual and Force Servoing in an Unknown Environment" *IEEE Robotics and Automation Magazine* December 1998.

[15] Xiao D, Gosh B, Xi N, Tarn TJ, "Integration of Real-Time Planning and Control in an Unstructured Workspace" In *Proc. of IROS* 1998, p1485-1490

[16] Garcia, Zangwill *Pathways to solutions, fixed points, and equilibria*, Prentice-Hall, 1981.

[17] Fletcher R. *Practical Methods of Optimization* Chichester, second ed. 1987

[18] Dahlquist G. Björck Å. *Numerical Methods* Second Ed, Prentice Hall, 199x, preprint.