Microrobotic Crystal Mounting Using Computer Vision

Atanas Georgiev[1], Peter K. Allen[1] and Youcef Mezouar[2]

[1] Department of Computer Science, Columbia University, New York, NY

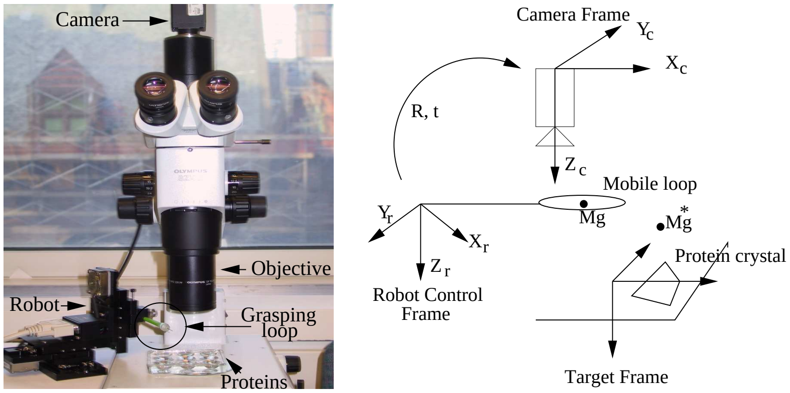
[2] LASMEA, Universite Blaise Pascal, Clermont-Ferrand Cedex, France

Abstract. In this paper, we present a framework for cell manipulation tasks with visual servoing micromanipulation strategies. We have designed a vision-based micropositioning system to address the requirement of high precision needed to perform manipulation of objects under 100 μm in size. We also propose designs for tooltips to use on the micromanipulator to effect an automated crystal mounting. The system calibration (microscope-camera-micropositioner) and the model of the observed scene are not known. Experimental results for micropositioning tasks with respect to protein cells are presented and demonstrate the validity of the proposed approach.

1 Introduction

Crystallized proteins are an important source for research in the biological sciences. They can be used in light-scattering experiments, growth-rate measurements, calorimetry, and the evaluation of new crystallization techniques and reagents, all of which contribute to a deeper understanding of their molecular structure. This will impact a number of fields, including the emerging structural genomics field [12], structure-directed drug design [14], and the newly developed screening by X-ray crystallography [11], as well as small molecule applications. Protein crystals are utilized for downstream processing and purification in the industrial production of enzymes, they are involved in pharmaceutical formulation, and they are often useful in various other areas of biophysical experimentation. The development of strategies and devices for automated and fast manipulation of protein crystals can greatly improve the throughput in molecular research by reducing from hours to seconds the time for manipulation of crystals.

This work is aimed at using computer vision to provide the compliance and the robustness which precise protein manipulation requires without the need for extensive analysis of the physics of grasping or a detailed knowledge of the environment. One of the major advances in robotics over the last 20 years is the visual control of robotic manipulators [7]. The advent of fast and inexpensive digital imaging technology has allowed camera systems to be integrated as part of a closed loop feedback control system [1]. Visual servoing strategies had been successfully implemented at the microscale level for manipulation of known micro-electromechanical systems [3, 10] with calibrated devices (however in these papers the CAD models of the proteins crystals and the system calibration are not known). Vision can provide rich knowledge about the spatial arrangement of the objects to be manipulated, as well as knowledge about the means of manipulation, which in our case are the instruments needed to perform



(a) (b)

Figure 1: Setup: (a) workstation for protein crystals manipulation; (b) workstation frames.

protein manipulation. Our goal is to visually monitor and control these instruments as they isolate and acquire proteins (a task known as *crystal mounting*). Using a vision system, a simple uninstrumented tool can become a precision device capable of position control. For tasks, such as cell or protein manipulation, this idea of visual feedback control becomes extremely important. Classical strategies of manipulation will not work at these scales (objects can be under $100\,\mu$m in size) due to the required precision (beyond the calibration range of conventional industrial precision devices) and additional problems relating to microscale phenomena. Currently, the mechanics of micromanipulation is poorly understood, and thus results of sensorless micromanipulation strategies are unpredictable.

This paper is based on our previous work [9] in which we proposed an integrated control system, consisting of a high-resolution optical microscope, digital imaging system, image based servo-controllers and a micromanipulator for precise positioning of a cryogenic loop with respect to a protein. In this paper we extend our previous work with a better mechanism for visual tracking of the loop. We also describe the problems associated with using loops for automated mounting and present design and implementation ideas for two different mounting tools which will facilitate the task and make the process more robust.

The remainder of this paper is organized as follows: In Section 2, we briefly describe our hardware setup and the task at hand. The next two sections summarize our previous work — Section 3 presents our method for bringing the loop into focus and Section 4 explains our visual servoing method. Next, in Section 5, we address some drawbacks of our original tracking method and present an improved version. Our work on methods for protein crystal mounting is presented in Section 6.

2 Setup and Task Description

Our workstation (see Figure 1a) is centered around a Sutter MP-285 micropositioner platform with three degrees of freedom — independent motion control along the X, Y and Z axes. This positioner allows sub-micron resolution down to $40\,nm$. A crystal mounting tool (e.g. a cryogenic loop, see Figure 2a) is attached as the manipulator

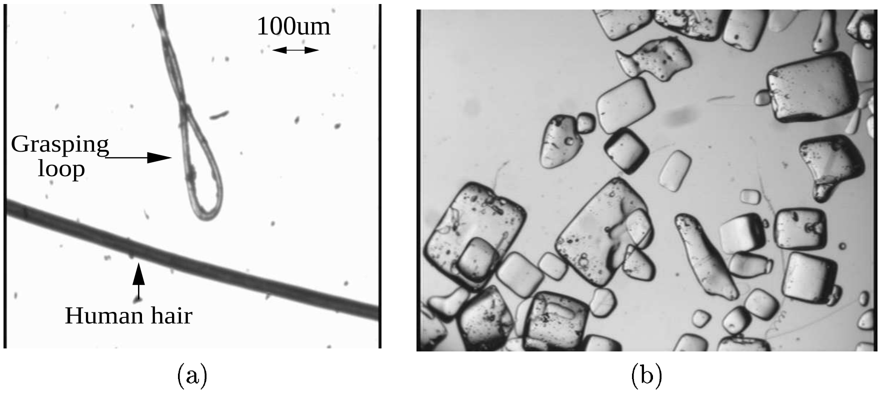


Figure 2: (a) Grasping loop for protein crystals; (b) protein crystals.

end-effector. The mounting tool is especially designed to pick protein crystals (Figure 2b). The tool and the protein crystals are observed through an Olympus SZX12 optical microscope. The microscope provides a total magnification from 8.4x to 108x. The unit has a CCD camera module adapter onto which a Sony XC-77 CCD camera is mounted.

In our system, the camera mounted on the microscope objective observes the protein crystals and the mounting tool (e.g. the cryogenic loop) attached to the three-degrees-of-freedom micropositioner (eye-to-hand configuration). Let $\mathcal{F}_c$ and $\mathcal{F}_r$ be respectively a frame attached to the fixed camera and a frame attached to the robot (see Figure 1b). The Z-axis of the camera frame and of the robot control frame are parallel.

Our goal is to align a reference point of the tool (e.g. the center of the hole for a loop) $\mathcal{M}$ with coordinates $\mathbf{M}_c = [X_c, Y_c, Z_c]^T$ in $\mathcal{F}_c$ and the center of an isolated protein $\mathcal{M}^*$ with coordinates $\mathbf{X}_c^* = [X_c^*, Y_c^*, Z_c^*]^T$ in $\mathcal{F}_c$. The protein is assumed in focus. Three degrees of freedom must be controlled, we need thus to extract at least three independent measures from the image. Since the protein is assumed in focus, $Z_c = Z_c^*$ when the loop is also in focus. As, we will see in the sequel, the Z-axis can thus be controlled using a focus measure T_r in the image.

Let $\mathbf{x} = [x, y]^T$ (resp. $\mathbf{x}^* = [x^*, y^*]^T$) be the coordinates in the image plane of the reference point of the mounting tool (resp. of the protein). The robot motion along the X and Y axis will be controlled by minimizing the distance in the image space between $\mathbf{x}$ and $\mathbf{x}^*$. The positioning task can thus be achieved by first minimizing a focus measure and then by minimizing an error in the image space. To control the X and Y axis of the micropositioner we have to track the tool in a complex scene.

3 Bringing the Loop into Focus

The first step of our control strategy is to bring the loop in focus. To realize the control along the Z-axis, the small depth of field of the microscope can be used. Since the target (i.e the protein crystal) is assumed in focus, the positioning of the micromanipulator along Z can be done by moving along the Z axis until a measure of focus in the image I is maximized. We chose as focus measure the Tenengrad measure

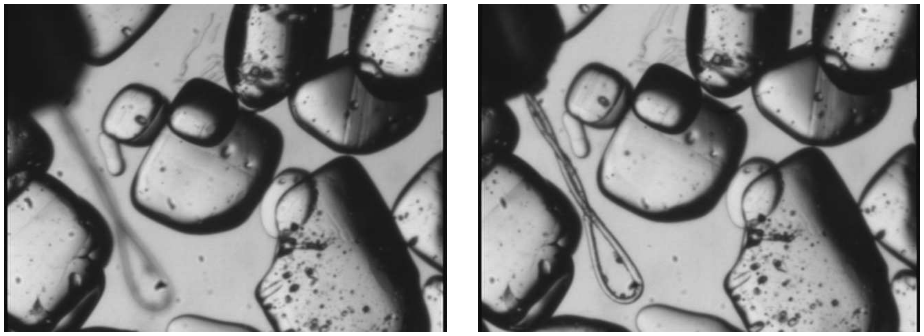


Figure 3: Focusing the loop: (a) initial defocused image; (b) final focused image.

T_r. Tenengrad is a measure of thresholded gradient magnitude [13]:

$$\left\{ \begin{array}{lcl} T_r & = & \sum_x \sum_y \|\vec{\delta I}\|^2 \\ & = & \sum_x \sum_y (\delta I_x^2 + \delta I_y^2) \end{array} \right. \quad \text{and } \|\vec{\delta I}\|^2 > \epsilon \tag{1}$$

It can be proved that the best focused image is obtained at the global maximum of the focus measure T_r if the threshold is zero [13]. We thus use the Tenengrad measure with $\epsilon = 0$. We maximize it by controlling the Z motion of the manipulator according to:

$${}^rV_z = \Gamma \dot{T}_r \tag{2}$$

where ${}^rV_z$ is the Z control expressed in the robot control frame and Γ depends on a number of parameters such as the refractive index of the optics, the magnification factor and others (see [9] for details). The norm of Γ can be fixed as a gain of a proportional control law and the sign of Γ ensuring the maximization of T_r can be obtained off-line by observing the effect of the robot Z motion on T_r.

This method has been tested on our experimental platform (see Figure 1). The specified visual task was to automatically bring the loop into focus from an initial out-of-focus position. The results are shown in the figures below. The initial defocused image is shown in Figure 3a and the in-focus image is shown in Figure 3b.

4 Visual Servoing

After the loop is focused, the motion in X and Y has to be controlled in order to align the center of gravity of the loop with the center of a protein crystal. The idea is to use visual servoing to slide the loop under the crystal and then lift it up with a simple upward motion.

Visual servoing is classified into two main approaches [6, 7, 15]. The first one [8, 16], based on the computation of a 3-D Cartesian error, requires a perfect CAD-model of the object and a calibrated camera to obtain unbiased pose estimation. In the second approach, the pose estimation is omitted and the control loop is directly closed in the image space. That ensures local convergence and stability in presence of modeling errors and noise perturbations [2]. In our case, the models of the observed targets are unknown and the system is not calibrated. We thus use the image-based approach.

Central to this approach is the image Jacobian $\mathbf{L}$ (also called interaction matrix) which relates the differential motion $\mathbf{ds}$ of some image features $\mathbf{s}$ to the differential motion in the camera coordinates $\mathbf{dP}$: $\mathbf{ds} = \mathbf{LdP}$. In our application, the camera observes the robot to be controlled; such configuration is known as eye-to-hand systems [4] as opposed to the more classical eye-in-hand configuration, where the camera is mounted on the robot effector and observes the object to be manipulated.

For our specific application, considering only a planar horizontal motion, the eye-to-hand Jacobian relationship for a single feature point is:

$$\dot{\mathbf{x}} = \mathbf{L}_{xy}\mathbf{R}_{xy}\left[\begin{array}{c} {}^{r}V_x \\ {}^{r}Vy \end{array}\right] = \mathbf{L}_{\mathbf{t}xy}\,{}^{r}\mathbf{V}_{xy} \qquad (3)$$

where $\mathbf{L}_{\mathbf{t}xy} = \alpha\mathbf{R}_{xy}$ is the eye to hand interaction matrix, α is the magnification factor, and $\mathbf{R}_{xy}$ is a 2×2 rotation matrix that maps the control vector expressed in the camera frame ${}^{c}\mathbf{V}_{xy}$ and the control vector expressed in the robot control frame ${}^{r}\mathbf{V}_{xy}$. A suitable control law to make the error vector $\mathbf{e} = \mathbf{x} - \mathbf{x}^*$ decrease exponentially (i.e, $\dot{\mathbf{e}} = -\beta\mathbf{e}$) to $\mathbf{0}$ is given by:

$${}^{r}\mathbf{V}_{xy} = -\beta\widehat{\mathbf{L}}^{+}_{\mathbf{t}xy}(\mathbf{x} - \mathbf{x}^*) \qquad (4)$$

where $\mathbf{x}^*$ is the desired position in the image space of the center of the loop, $\widehat{\mathbf{L}}_{\mathbf{t}xy}$ is the estimated image Jacobian and β is a proportional gain.

Judging from a stability study, we fix $\widehat{\mathbf{L}}_{xy} = \widehat{\alpha}\widehat{\mathbf{R}}$ to a constant value while ensuring the convergence of control. For our application, $\widehat{\alpha}$ is fixed to the value given by the manufacturer. The estimated rotation matrix $\widehat{\mathbf{R}}_{xy}$ is fixed as the closest rotation matrix to $\frac{1}{\alpha}\widehat{\mathbf{L}}^{0}_{xy}$ where $\widehat{\mathbf{L}}^{0}_{xy}$ is the interaction matrix numerically computed off-line by observing the repercussion of the robot motion in the image space. Further details can be found in [9].

To track the center of gravity of the grasping loop, we initially used a simple motion detector. Given two consecutive images, I_{t-1} and I_t, we threshold their difference to obtain a binary image of temporal variations $I_t - I_{t-1}$. Similarly, we use a reference image I_0 (say, the initial image) to obtain $I_t - I_0$. The logical operation AND is then applied to the two binary images to obtain the loop and its center is computed as the center of gravity of its inner pixels.

We have used this tracking method to verify our servoing experimentally on our platform. The specified visual task consisted of precise positioning of the cryogenic loop with respect to a crystal protein (after the focusing task was achieved). The grasping loop was visually guided to its desired position under the described method. The images corresponding to the desired and final configurations are given in Figures 4a and 4b respectively (the initial and desired position of the center of gravity of the grasping loop are represented by white disks). Figure 4c shows the error in the coordinates of the center of gravity of the grasping loop between its current and its desired location in the image. The convergence of the coordinates to their desired value demonstrates the correct realization of the task. The final errors on the coordinates is less than 1 pixel that corresponds to a positioning accuracy beyond $10^{-1}\mu m$. The computed control law is given in Figure 4d.

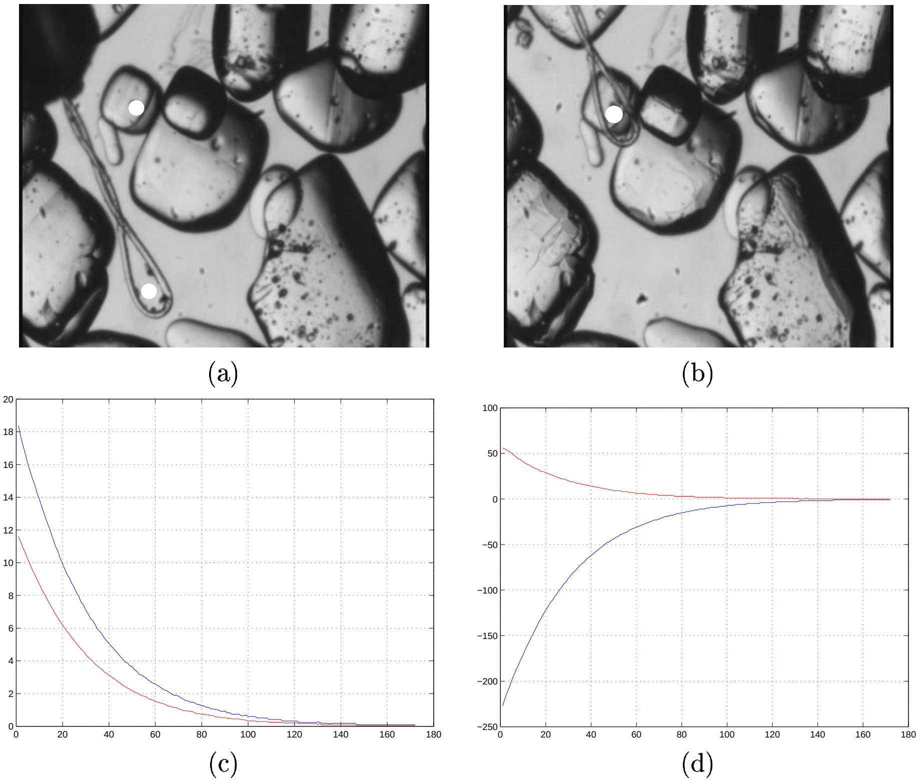


Figure 4: (a) Initial image; (b) final image; (c) control vector ($\mu m/s$); (d) error in the image space (pixel versus time).

5 Tracking

One of the problems we encountered with our original tracking method is the assumption that the only object in motion is the tooltip (e.g the loop). In practice, this assumption leads to problems. Due to the nature and the size of the environment ($2-3\mu l$ of liquid), the motion of the tool stirs and causes the other nearby objects to move. This also causes variation in the scene illumination. The net effect is that the tracker may sometimes get confused. Additionally, for a successful mount, we need to track the target crystal, whether moving or not.

To resolve these complications, we have employed an alternative method to track the location of the loop based on SSD template matching [5]. We also use template matching to track the target protein crystal.

For the crystal, given two consecutive images I_t and I_{t+dt} taken at times t and $t+dt$, we use the sum of squared differences as a measure of quality of the match between position $[x,y]^T$ of the crystal in image I_t and position $[x+dx, y+dy]^T$ in image I_{t+dt}:

$$F(x,y,x+dx,y+dy) = \sum_{(i,j)\in\mathcal{R}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i,y+j))^2 \quad (5)$$

Here, $\mathcal{R}$ is a rectangular region of interest of width w and height h centered at $[0,0]^T$ which is used to locate the protein crystal. The new location of the crystal $[x^*, y^*]^T$, is

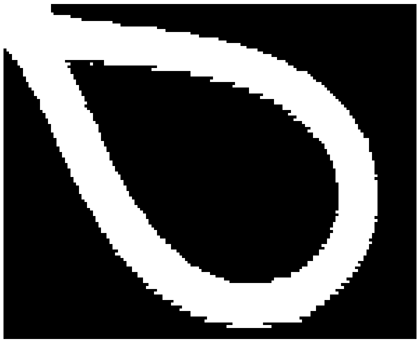


Figure 5: A sample loop mask.

the one that minimizes $F(x, y, x+dx, y+dy)$ for $[dx, dy]^T$ in an $N \times N$ neighborhood of $[0, 0]^T$.

For the loop, however, the above SSD tracking will not work well, since the area of the floss is quite small compared to the area of the rectangle that encompasses the loop, exposing more of the background than the loop itself. Generally, if the background is more or less uniform and the loop is highly contrasting, the chance of the tracker missing the loop will be quite low. However these assumptions do not always hold true and there are scenarios under which the tracker will fail (e.g. see the Appendix).

In order to avoid this situation, we have to make sure that we use as a template only pixels that we know belong to the loop and avoid background pixels. To do so, we use a binary mask that will filter out the background. The mask (Figure 5) is a simple binary matrix $K_{i,j}$ of size $w \times h$, which is registered with the tracking window and contains ones, where the corresponding pixel belongs to the loop, and zeros, where the pixels are part of the background. Thus we minimize the function

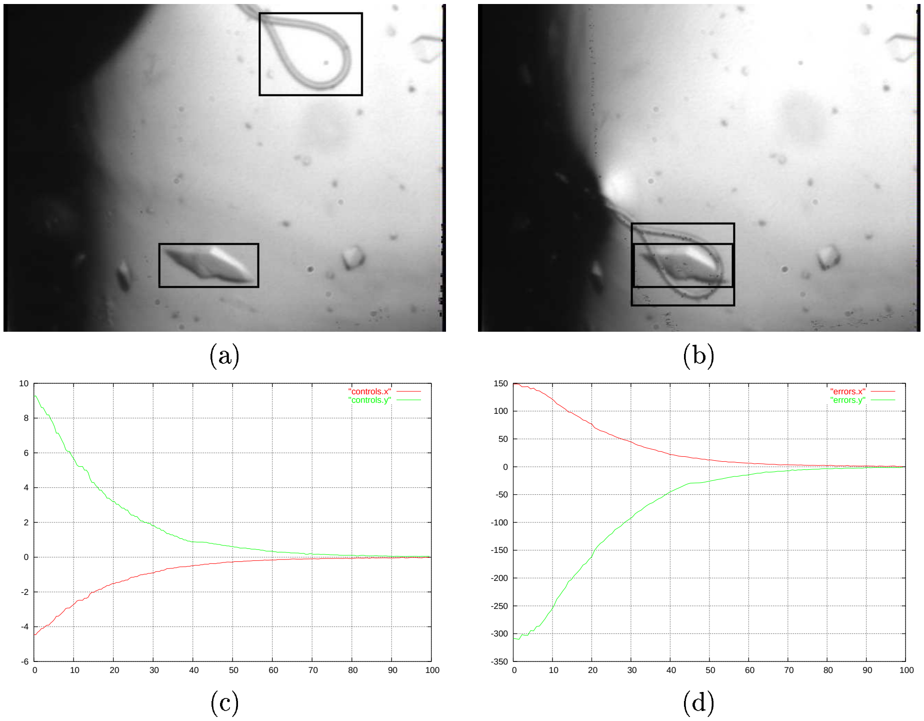
$$\hat{F}(x, y, x+dx, y+dy) = \sum_{(i,j)\in\mathcal{R}} K_{i,j}(I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2 \quad (6)$$

The mask is initialized following the technique described in Section 5 by controlling the manipulator in the beginning to go to three different locations and acquiring three initial images, I_0, I_1 and I_2. The loop is detected on I_2 and a rectangle is selected around the loop which is used as the mask (Figure 5). The corresponding area of I_2 is used to initialize the SSD tracking algorithm.

We have tested this tracking method experimentally using the same test as described in Section 5. An experimental run is shown in Figure 6. The larger rectangle on the images shows the tracking window at the beginning and at the end of the run. The smaller rectangle tracks the crystal. As can be seen from the figure, the new method successfully tracks the loop and converges to the target. The final errors on the coordinates is again less than $10^{-1}\mu m$.

6 Protein Crystal Mounting

The idea behind aligning the center of the loop with the center of the selected protein is that on its way to the target point, the loop will slide under the crystal and then


(a)

(b)

(c)

(d)

Figure 6: (a) Initial image; (b) final image; (c) control vector ($\mu m/s$); (d) error in the image space (pixel versus time).

the manipulator could simply move up to lift it. This is certainly possible, as our experience shows. However, it is not very robust in practice even if the trackers perform flawlessly. Two other things may happen instead.

First, during crystallization, some proteins tend to attach themselves to the bottom as if glued. This is not rare but a perfectly normal phenomenon. If the loop encounters such a protein, it will most likely bend and flex because it is much softer than the crystal. In such situations, crystallographers first use a sharp solid tool and try to detach the crystal before they proceed with the loop. This is a very delicate procedure because it is extremely important that the crystals remain intact and they are typically very fragile. Our micromanipulator, however, does not allow for automatic change of the tools, so this solution is not viable.

Second, even if the crystal is loose, the loop may simply start pushing it, instead of sliding underneath. This is often the case with smaller crystals as they are very light and their height is comparable to the thickness of the floss. Crystallographers address the situation by using rotational degrees of freedom to perform a "scooping" move and unearth the crystal with the loop. Our manipulator does not provide these degrees of freedom. Even if it did, it is not clear that the same move can be easily imitated programmatically.

To address these two issues, we have chosen two alternative tools to replace the loop — a shovel and a glass capillary. The shovel (Figure 7a) is made of a small rectangular piece, approximately $1mm \times 0.3mm$ in size, of plastic shim stock glued to the tip of a pin or copper wire. It operates on the same principle as the loop — we slide it under the crystal — however, it is $25\mu m$ thick which makes it hard enough to

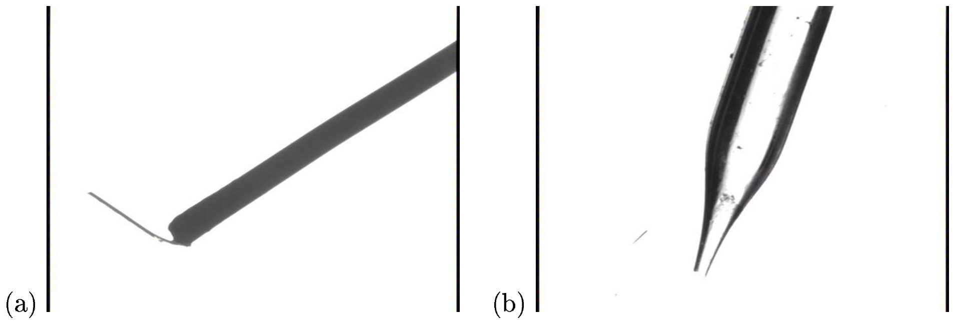


Figure 7: (a) The shovel design; (b) a glass capillary.

exercise the necessary force to “unglue” stuck crystals. It has also much larger area than the loop which stirs the liquid around it as it moves. This prevents the nearby crystals from lying on the bottom and makes it easier for the shovel to go underneath them.

The glass capillary (Figure 7b) is not a new idea. In fact, it was what crystallographers used for mounting before they switched to loops (for reasons having to do with ease of manual use and postprocessing). It is very useful in that it can easily pick a nearby crystal if a small negative pressure is applied to its other end. Its tip is hard enough to help loosen glued crystals.

The capillary is mostly phased out of use in favor of the loop these days because it makes it hard to properly freeze the crystal before it is placed on the x-ray diffractometers. However, a very useful feature of a capillary is that it can easily eject the crystal back by applying a small positive pressure. This opens the possibility for a two-step crystal mount scenario, involving a transfer station as opposed to the direct deposit method with the shovel: the capillary could be used to acquire the crystal from the liquid and then deposit it to the loop in a much more controlled fashion.

To test the usability of the two tools we have performed real-world experiments. The goal of each experiment was to only confirm that the tool is capable of picking a crystal. Each tool, was installed on the manipulator which was teleoperated to perform a number of mounting tasks. Figure 8 shows a select images from a successful mounting onto a shovel. Figure 9 shows the use of a capillary to pick and later release a crystal. The advantages of both tools compared to using a loop were clearly experienced due to the ease of the crystal manipulation.

7 Conclusion

The development of visually guided strategies applied to high resolution optical systems show promise in overcoming a technology barrier to the automated micromanipulation of biological cells. In this paper, we have investigated and experimentally validated the use of uncalibrated visual servoing techniques to position a loop mounted on a micromanipulator with respect to a protein crystal observed by a camera mounted on an optical microscope. The experimental results show that the visual feedback can provide the required precision to achieve the fixed task. We proposed designs for tools that make the tracking and mounting task easier and more robust.

This work is our next step forward on the design of an integrated system that will be able to visually isolate individual proteins in a culture, to move customs

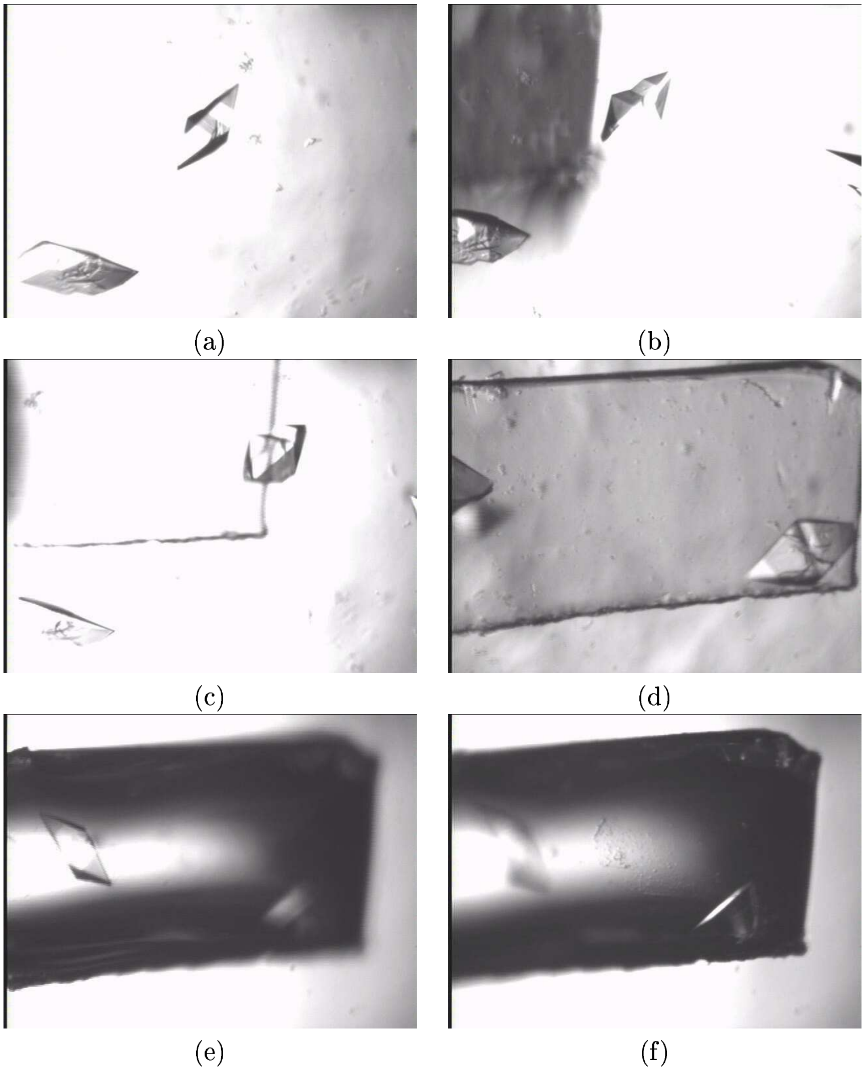
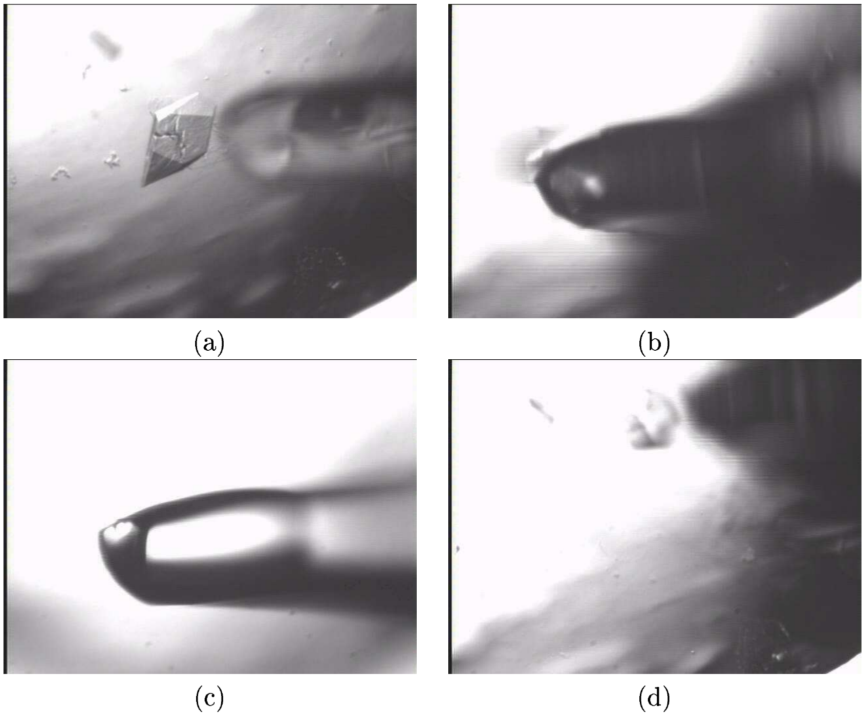


Figure 8: A successful mount sequence using the shovel: (a) initial crystal configuration; (b) the shovel is entering the liquid (dark patch on the left); (c) the shovel slides under one crystal; (d) the shovel slides under a second crystal (part of the first one is visible on the left); (e) shovel raised above the solution, first crystal in focus (f) shovel raised above the solution, second crystal in focus.



(a) (b)

(c) (d)

Figure 9: A successful crystal picking sequence using the capillary: (a) the capillary approaching from the right; (b) the capillary sucks in the crystal; (c) the capillary above the liquid with the crystal at the tip; (d) the capillary releases the crystal back in the liquid.

instruments to contact the isolated proteins, to grasp the proteins and transport them for further processing.

Acknowledgement

This work was supported by a grand to the Northeast Structural Genomics Consortium from the Protein Structure Initiative of National Institutes of Health (P50 GM62413).

Appendix: Sensitivity of the Tracking Algorithm

One of the problems with tracking the loop is that a selected rectangular template will inevitably include a portion of the background which may affect the tracker. The standard SSD tracking of a rectangular region will not work well, since the area of the floss is quite small compared to the area of the rectangle that encompasses the loop, exposing more of the background than the loop itself.

Specifically, assume we have a rectangular region of interest $\mathcal{R}$ around the loop of size $w \times h$ and let the ratio of the area of the loop in that rectangle to the total area of $\mathcal{R}$ is k, i.e. the loop occupies $k\,w\,h$ pixels and the background occupies $(1-k)\,w\,h$ pixels, $0 \leq k \leq 1$. Let the loop be at position $[x,y]^T$ at time t and occupy the subset of pixels $\mathcal{L}_t \subset \mathcal{R}$. Similarly, let the loop at time $t+dt$ be at position $[x+dx, y+dy]^T$ and occupy the area $\mathcal{L}_{t+dt} \subset \mathcal{R}$. Let $\mathcal{L} = \mathcal{L}_t \cup \mathcal{L}_{t+dt}$ and $\mathcal{B} = \mathcal{R} \setminus \mathcal{L}$. Now consider the two values $F(x,y,x+dx,y+dy)$ and $F(x,y,x,y)$.

For $F(x,y,x+dx,y+dy)$, the region of interest has moved by the same amount $[dx,dy]^T$ as the loop and therefore the loop occupies the same portion of $\mathcal{R}$, i.e. $\mathcal{L}_t = \mathcal{L}_{t+dt} = \mathcal{L}$. This is the desired situation when the tracker has successfully followed the loop. Therefore,

$$F(x,y,x+dx,y+dy) = \tag{7}$$

$$\sum_{(i,j)\in\mathcal{R}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2 \tag{8}$$

$$= \sum_{(i,j)\in\mathcal{L}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2 + \tag{9}$$

$$\sum_{(i,j)\in\mathcal{B}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2 \tag{10}$$

$$\approx \sum_{(i,j)\in\mathcal{B}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2 \tag{11}$$

$$\approx \sum_{(i,j)\in\mathcal{B}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2 \tag{12}$$

$$\approx (1-k)\,w\,h\,\frac{\sum_{(i,j)\in\mathcal{B}} (I_{t+dt}(x+dx+i, y+dy+j) - I_t(x+i, y+j))^2}{(1-k)\,w\,h} \tag{13}$$

$$\approx (1-k)\,w\,h \times MSE(\mathcal{B}) \tag{14}$$

where $MSE(\mathcal{B})$ is the mean-square-error of the intensity of the background pixels. The sum over the region $\mathcal{L}$ will ideally be zero, because it takes the difference of the intensities of the same part of the loop between the two images. In practice it will be a small, positive amount due to imperfections of the acquisition device.

Evaluating $F(x, y, x, y)$ corresponds to the situation where the tracker has locked onto the background. In this case, the region that undergoes changes consist of the old and the new positions of the loop in $\mathcal{R}$ and the rest of it is the background $\mathcal{B}$ which stays the same (subject to repeatability characteristics of the acquisition device):

$$\begin{aligned}
F(x,y,x,y) = & \qquad (15)\\
\sum_{(i,j)\in\mathcal{R}} (I_{t+dt}(x+i, y+j) - I_t(x+i, y+j))^2 & \qquad (16)\\
= \sum_{(i,j)\in\mathcal{L}} (I_{t+dt}(x+i, y+j) - I_t(x+i, y+j))^2 + & \qquad (17)\\
\sum_{(i,j)\in\mathcal{B}} (I_{t+dt}(x+i, y+j) - I_t(x+i, y+j))^2 & \qquad (18)\\
\approx \sum_{(i,j)\in\mathcal{L}} (I_{t+dt}(x+i, y+j) - I_t(x+i, y+j))^2 & \qquad (19)\\
\approx 2\,k\,w\,h \, \frac{\sum_{(i,j)\in\mathcal{L}} (I_{t+dt}(x+i, y+j) - I_t(x+i, y+j))^2}{2\,k\,w\,h} & \qquad (20)\\
<= 2\,k\,w\,h \times MSE(\mathcal{B}, \mathcal{L}) & \qquad (21)
\end{aligned}$$

where $MSE(\mathcal{B}, \mathcal{L})$ is the mean-square-error of the intensity difference of corresponding pixels that change between the loop and the background. The inequality at the end follows from the fact that the area of $\mathcal{L}$ is at most twice the area of $\mathcal{L}_t$.

Since we want to track the loop, we want to make sure that $F(x, y, x+dx, y+dy)$ minimizes F in the search neighborhood, which means that it should be less than $F(x, y, x, y)$. However, this will not be the case when:

$$F(x,y,x,y) <= 2\,k\,w\,h\,MSE(\mathcal{B},\mathcal{L}) < (1-k)\,w\,h\,MSE(\mathcal{B}) \qquad (22)$$

or

$$MSE(\mathcal{B},\mathcal{L}) < \frac{(1-k)}{2\,k}\,MSE(\mathcal{B}) \qquad (23)$$

To give real-life meaning of this, if the loop occupies $1/5$th of the area of the rectangle and the square difference on the average between the loop and the background is about 100 intensity levels, then a background with intensity variance of 50 levels or more may cause the tracker to lose the loop.

In order to avoid this situation, we have to make sure that the inequality (23) is never true. The solution is to make the coefficient $\frac{(1-k)}{2\,k}$ as small as possible, i.e. k should be as close as possible to 1. This means that we should strive to restrict the template only to pixels that we know belong to the loop and avoid the inclusion of as many background pixels as possible.

References

[1] P. K. Allen, A. Timcenko, Y. B., and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. on Robotics and Automation*, 9(2):152–165, April 1993.

[2] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control, D. Kriegman, G. Hager, A. Morse (eds), LNCIS Series, Springer Verlag*, 237:66–78, 1998.

[3] J. T. Feddema and T. R. Christenson. Parallel assembly of high aspect ratio microstructures. In *SPIE Conference on Microrobotics and Micromanipulation*, pages 153–164, Boston, USA, September 1999.

[4] G. Flandin, F. Chaumette, and E. Marchand. Eye-in-hand / eye-to-hand cooperation for visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, April 2000.

[5] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, October 1998.

[6] K. Hashimoto. *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback.* World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.

[7] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[8] P. Martinet, N. Daucher, J. Gallice, and M. Dhome. *Robot control using monocular pose estimation.* Worshop on New Trends in Image-based Robot Servoing, IROS'97, Grenoble, 1997.

[9] Y. Mezouar and P. K. Allen. Visual servoed micropositioning for protein manipulation. In *Int. Conf. Intelligent Robots and Systems, Lausanne, Switzerland*, pages 1766–1771, 2002.

[10] B. J. Nelson, Y. Zhou, and B. Vikramaditya. Sensor-based microassembly of hybrid mems devices. *IEEE J. of Control System*, pages 35–45, December 1998.

[11] V. L. Niennaber, P. Richardson, V. Klighover, J. Bouska, V. L. Giranda, and J. Geer. Discovering novel ligands for macromolecules using x-ray crystallographic screening. *Nat. Biotechnol.*, 18:1105–1108, 2000.

[12] R. F. Service. The automated approach to protein structure. *Science 285*, 1345, 1999.

[13] M. Subbarao, T. Choi, and A. Nikzad. Focusing techniques. Technical Report 92.09.04, Dept. of E.E, SUNY Stony Brook, 1992.

[14] C. L. Verlinde. Structure-based drug design: progress, results and challenges. *Structure 2*, pages 577–587, 1994.

[15] L. Weiss, A. Sanderson, and C. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, October 1987.

[16] W. Wilson, C. W. Hulls, and G. Bell. Relative end-effector control using cartesian position-based visual servoing. *IEEE Trans. on Robotics and Automation*, 12(5):684–696, 1996.