# Robot Stereo-hand Coordination for Grasping Curved Parts

Yves Dufournaud, Radu Horaud, Long Quan

# Robot Stereo-hand Coordination for Grasping Curved Parts

Yves Dufournaud Radu Horaud Long Quan
first.last@inrialpes.fr
GRAVIR–IMAG & INRIA Rhône-Alpes
655, avenue de l'Europe
38330 Montbonnot Saint Martin, FRANCE

**Abstract**

In this paper we present an algorithm to compute set-point (i.e. a position to be reached by the robot) automatically from conic features virtually placed by an operator onto the object. We then use a visual servoing algorithm to guide the gripper to its final position. We have tested this algorithm in trying to open a valve with a 6 degree of freedom robot arm, using only visual information and without any model of the valve.

keywords: conic pose, hand-eye coordination, visual servoing.

## 1 Introduction

Grasping is one of the most common tasks in robotics and telemanipulation. But it is also one of the most difficult to accomplish when operating on objects of unknown shape and size in hazardous or remote environments. Although this task is crucial for many applications (space exploration, nuclear plants inspection, etc.), few systems are able to operate with sufficient flexibility and reliability in complex situations. The difficulties are twofold.

First, to grasp an object implies to select a grasp (a relationship between the gripper and the object) and to represent this relationship in some space. Building this relationship requires a gripper and an object model. But often the latter is unavailable or poorly known.

Second, moving the gripper to the final grasp location is a difficult task by itself, since robot-hand motion may be subject to various perturbations: robot kinematics are only partially known, unpredictable obstacles may be present, and the grasp position can be unprecise.

In the past, a number of sensors have been used to control grasping, including proximity and tactile sensors. For various reasons none are fully satisfactory. The former type requires that the gripper and object lie within one centimeter while the latter requires that gripper and object touch each other. Using vision to control grasping seems natural, since it allows to recognize and to locate objects, and to track the gripper to a desired position with motion control.

Stereo vision has already been used in conjunction with robot control by a number of authors [6] [4]. But these approaches use stereo to construct points and lines whereas most objects are curved.

Our approach to perform grasping is to consider a stereo rig mounted onto a fixture which provides visual feedback to control the end-effector of a manipulator. The location of the stereo rig with respect to the robot is unknown, as well as the location of the object with respect to the robot and the stereo rig. The shape and size of the object itself are unknown, but we assume that parts to be graspped can be fitted in a rotationally symmetrical form.

The main contribution of this paper is to demonstrate how recent results in computer vision can be used and integrated to build a system able to manipulate unknown objects in complex environments. Previous work on visual servoing required a model of the object being manipulated to be known in advance [1]. This model is needed because grasp computation involves representing a 3-D relationship between the gripper and the object. A key idea in order to manipulate objects with *a priori* unknown models is to find a feature model that is sufficiently simple to be fitted to a numerous range of objects, and sufficiently elaborate to permit efficient localization and manipulation, i.e. has implicit semantic.

We find that conics can be one of these features. Indeed a lot of objects which need to be manipulated and grasped contain curved and specifically rotationally symmetrical parts. These parts can be used for grasping by fitting an ellipse to them. We take as an example the case of a valve to be opened by a robot arm, and show that such a feature can be successfully used in this task.

The next section summarizes our approach of the overall process of visual guided grasping. Section 3 shows how to use handle fitting to perform grasping on unknown objects with circular shape. Section 4 gives some details on the algorithm used to compute pose with conics. Section 5 shows some results obtained and gives directions for future work.

## 2 Overview of Image-based Grasping

Here, we give an overview of our approach for grasping, and explain our choices. What we mean by a grasp is in fact only a specific position of the gripper with respect to some physical object, and grasping an object is actualy controling the gripper to this position.

This gripper-object relationship is defined in 3-D space, and is usually represented by a transformation from the object frame to the gripper frame. Alternatively any gripper-object alignment, or any object-to-object relationship can be represented in terms of relationship between points detected on the objects (Fig. 1). The choice of these points depends on the visual sensor being used, and hence upon feature extraction. The 3-D space used can be projective space [7], but in this case the gripper-object relationship must be learned prior to any grasp process. As we want to avoid such learning, which requires a human operator to put the gripper in the grasp position, we are restricted to Euclidean space.

Once this gripper-object relationship has been defined, it can be used, provided that the object has been located with respect to the servoing frame. Using image-based servoing then implies that: first, the object pose must be computed, which gives us a camera-object relationship. This in conjonction with the gripper-object relationship allow us to project gripper points in the images to predict the grasp position. This visual prediction is used by the servoing algorithm as the goal to achieve in the images (Fig. 2). This approach

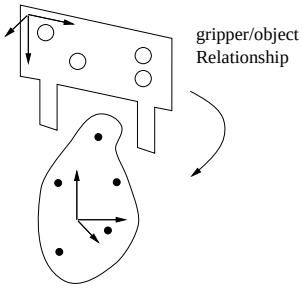


Figure 1: Grasp modelization using the transformation between the gripper frame and the object frame.

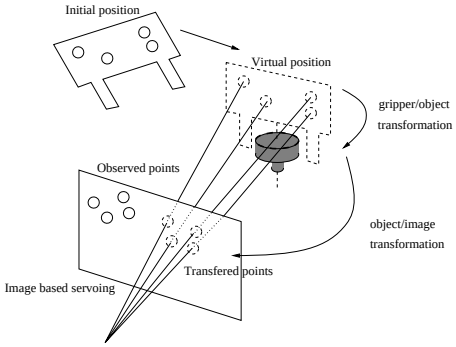


Figure 2: This figure shows the runtime setup where the gripper is visually servoed from an initial position. The goal position is defined by the image transfered points, and comes from a 3-D virtual position of the gripper.

has many benefits [1][7]. Indeed, the servoing algorithm is based on a closed control loop which gives a robust approach since we do not rely on robot frame coordinates to operate. As these coordinates are computed with joint information, which could be quite inaccurate, we avoid such errors. Second, as we use visual information to control the movement – by minimizing the error between the predicted target positions and the current location of the gripper (Fig. 8) – we avoid reconstruction errors as well (provided that our prediction of the final position is accurate). It is worth noting that visual servoing requires only one camera, and if we were allowing a CAD model of the object to be known, pose computation could be done with one camera also [1]. But as our goal is to grasp unknown objects, we need two cameras for reconstruction. Moreover as we need Euclidean reconstruction to build gripper-object relationship, we need a calibrated stereo rig. Nevertheless, experiments show that the pose algorithm used is robust [8], and [2] has recently shown that internal camera calibration weakly affects the convergence of image based robot control when only one camera is used.

The overall process can then be summarized in two stages both relying on visual information only:

- An off-line stage which corresponds to the grasp selection process. This involves low level image processing for salient edge extraction: two images of the scene are taken and processed. As our major concern is remote controlled robotics, we suppose at this stage that a human operator provides the knowledge to locate *interesting* features in both images. This is done by selecting edges corresponding to the contour of a specific part. In each image a conic is fitted to the selected edges. Once this selection and matching process is complete, pose computation is done (see section 4) with the assumption that each of the selected conic corresponds to the projection of a 3D conic lying in a space plane seen by the two cameras. After this the computed pose is used to predict a gripper position satisfying grasping in the two images (Fig. 7).

- An online stage in which the predicted position is used as a goal for the visual servoing algorithm (Fig. 8).

# 3 Set-point computation

The aim of set-point computation is twofold: it is used to define how to place the gripper for effective grasping, and to control the robot arm. From the latter it is clear that its representation will depend on the chosen control process. As we use visual servoing, set-point (i.e. a position to be reached by the robot) in our case is defined as a set of points in image.

Defining how to place the gripper involves building a relationship between the object and the gripper. Besides the operator choice, this relationship can be constrained in several ways. For example in our experiments, we put marks onto the gripper to reduce the computational complexity of the tracking. The backdrop is that these marks must stay visible during all the servoing process. Therefore, the grasp position computed must satisfy this constraint taking into account the location of the two cameras such that the visual servoing process is optimal [3]. Nevertheless, set-point computation can be a very complex task, particularly if motion planning and obstacle avoidance are needed. Here we restrict ourselves to the simpler case of final gripper position prediction.

## 3.1 Polyhedral case

One of the many advantages of polyhedral objects is that they can generally be handled easily by a simple gripper with parallel claws. Conceptually, to a particular gripper belongs a class of handles that the gripper can grasp. These concepts are dual. In the case of the widely used parallel claws gripper, the class of handles that best fit are boxes aligned with the claws (Fig. 3). So with a definition of a particular gripper of that kind (mainly its physical property concerning its opening) we can say for a particular handle, logically built on a physical object, if it can be taken and how.

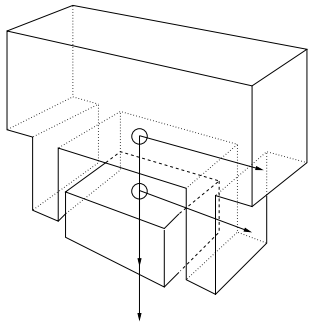

Figure 3: Handle and gripper modelization.

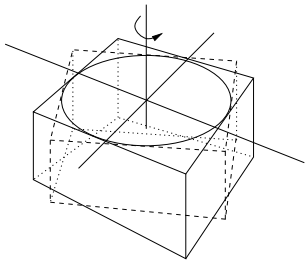

Figure 4: Cylindrical and elliptical case.

For polyhedral objects with a CAD model, one way of computing set-point for grasping is then to detect where such handles can be put on the model and if the gripper can

reach it with respect to its current location. This can lead to two processes: one generating exhaustively every possible handle on an object model (which can be pre-computed) and a second that filters these handles with situation constraints. For non-polyhedral objects, considering that the gripper properties remain, the difficulty is to fit the previous model onto actual objects.

### 3.2 The Ellipsis case

In the case of an object with elliptical parts, a box handle can be put by aligning ellipse axes with box faces (Fig. 4). This fully constrains the gripper position to 4 positions. The more interesting case is when the ellipse can be approximated by a circle. In the circle case there is one degree of freedom left: the orientation of the handle with respect to the circle axis. Maximum visibility of gripper marks (expressed as solid angle criteria) constrains this to one solution in each image. Using these solutions, a goal position is generated for each image (by computing final positions of the marks) and the best one (always is the sense of solid angle) is used to control the gripper in a monocular visual servoing process.

## 4 Pose computation

In this section we recall results on pose computation of conic from two views. Basically this is done by computing the intersection of the two projective cones built on the conic in each image (Fig. 5). The following shows how to compute this intersection.

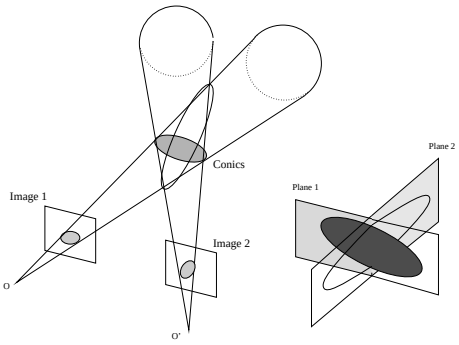


Figure 5: Projective cone intersection, with the two conic solutions and the degenerated 2-planes quadric represented.

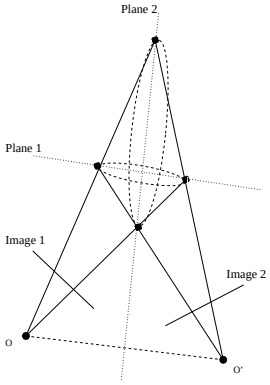


Figure 6: Plane selection: one plane is always between the camera centers, therefore it cannot be a solution in case of non-transparent objects.

### 4.1 Problem formulation

Let $\mathbf{x}$ be a point in space, and $\mathbf{u}$ and $\mathbf{u}'$ its projections in each image. These projections are each represented by a $3 \times 4$ matrix [5]:

$$\mathbf{u} = \mathbf{P}\mathbf{x} \text{ and } \mathbf{u}' = \mathbf{P}'\mathbf{x} \tag{1}$$

Assume that we have two images $C$ and $C'$ of a conic $\mathcal{C}$ lying somewhere in a space plane. $C$ and $C'$ are also conics and can each be represented by a $3 \times 3$ matrix so that:

$$C \equiv \mathbf{u}^t\mathbf{C}\mathbf{u} = 0 \text{ and } C' \equiv \mathbf{u}'^t\mathbf{C}'\mathbf{u}' = 0 \tag{2}$$

We are searching for the conic in space $\mathcal{C}$ which have been projected in $C$ and $C'$. As a conic is generally represented as the complete intersection of a quadric surface and a plane, we will reconstruct it this way by finding the plane in which it lies and intersecting it with one of the projective cones.

The projective cone $Q$ passing through a conic is obtained by substituting (1) in (2):

$$\mathbf{u}^t\mathbf{C}\mathbf{u} = (\mathbf{P}\mathbf{x})^t\mathbf{C}(\mathbf{P}\mathbf{x}) = \mathbf{x}^t(\mathbf{P}^t\mathbf{C}\mathbf{P})\mathbf{x} = 0$$

$$\text{That is: } \mathbf{x}^t\mathbf{Q}\mathbf{x} = 0, \text{ with } \mathbf{Q} = \mathbf{P}^t\mathbf{C}\mathbf{P}$$

$\mathbf{Q}$ is a $4 \times 4$ matrix of rank 3 (since rank$(\mathbf{P}) = 3$ and rank$(\mathbf{C}) = 3$) and symmetric (since $\mathbf{C}$ is symmetric). This ensures that $\mathbf{Q}$ is a proper cone [9]. Moreover $\ker(\mathbf{Q}) = \ker(\mathbf{P})$ which means that the vertex of the cone is the projection center of the camera represented by $\mathbf{P}$. So Q is the projective cone that link $C$ and $\mathcal{C}$.

Thus we obtain $Q$ and $Q'$ the projective cones which transforms $\mathcal{C}$ in $C$ and $C'$, respectively:

$$Q \equiv \mathbf{x}^t\mathbf{Q}\mathbf{x} = \mathbf{x}^t\mathbf{P}^t\mathbf{C}\mathbf{P}\mathbf{x} = 0$$

$$Q' \equiv \mathbf{x}^t\mathbf{Q}'\mathbf{x} = \mathbf{x}^t\mathbf{P}'^t\mathbf{C}'\mathbf{P}'\mathbf{x} = 0$$

### 4.2 Polynomial condition

To compute the intersection of these two cones we consider the pencil of quadrics $\mathbf{Q} + \lambda\mathbf{Q}' \equiv \mathbf{x}^t(\mathbf{Q}+\lambda\mathbf{Q}')\mathbf{x} = 0$. All quadrics of this family have in common the intersection of $Q$ and $Q'$. It is worth noting that the intersection of two cones in general is a quartic curve (i.e. an equation of degree 4). But as we made the assumption that $C$ and $C'$ are images of a conic, this conic is part of the intersection. As one component of the intersection (a quartic) is a conic (an equation of degree 2) the residual component is also a conic.

One special form of quadric is a pair of planes. This is a degenerate quadric of rank 2, which belongs to the previous quadric pencils [9]. We then restrict ourselves to search $\lambda$ so that $\mathcal{Q}(\lambda) = \mathbf{Q} + \lambda\mathbf{Q}'$ is a quadric of rank 2 (two planes). The initial conic $\mathcal{C}$ is the intersection of one of these two planes with the projective cones.

We do not list here all the results needed to justify the existence of this conic [8]. We just provide the necessary steps for computation of this pose.

Consider the characteristic polynomial of $\mathcal{Q}(\lambda)$:

$$|\mathcal{Q}(\lambda) - \mu\mathbf{I}| = \mu^4 + a_3(\lambda)\mu^3 + a_2(\lambda)\mu^2 + a_1(\lambda)\mu + a_0(\lambda) = 0 \tag{3}$$

In order to have $\mathcal{Q}(\lambda)$ of rank 2, it must have two distinct nonzero eigenvalues and a double zero eigenvalue. This implies $a_1(\lambda) = 0$ and $a_0(\lambda) \equiv |\mathcal{Q}(\lambda)| = 0$ therefore:

$$a_0(\lambda) \equiv |\mathcal{Q}(\lambda)| = |\mathbf{Q} + \lambda\mathbf{Q}'| = I_4\lambda^4 + I_3\lambda^3 + I_2\lambda^2 + I_1\lambda + I_0 = 0$$

where $I_j$ are polynomial coefficients obtained by calculating $|\mathcal{Q}(\lambda)|$. Since $\mathbf{Q}$ and $\mathbf{Q}'$ are of rank 3, then $I_4 \equiv |\mathbf{Q}| = 0$ and $I_0 \equiv |\mathbf{Q}'| = 0$. Thus $a_0(\lambda)$ could be written as:

$$a_0(\lambda) = \lambda\left(I_3\lambda^2 + I_2\lambda + I_1\right) = 0$$

To get a matrix of rank 2 from the pencil we must have two equal roots for this equation [8]. But two values are already used: $\lambda = 0$ and $\lambda = \infty$ which belong respectively to $a_0(0) \equiv |\mathcal{Q}(0)| = |\mathbf{Q}| = 0$ and $a_0(\infty) \equiv |\mathcal{Q}(\infty)| = |\mathbf{Q}'| = 0$. Thus we are only interested in finding the roots of:

$$I_3\lambda^2 + I_2\lambda + I_1 = 0$$

As the two roots are equal and different from 0 and $\infty$ , this second degree equation satisfies the condition:

$$\Delta \equiv I_2^2 - 4I_1I_3 = 0 \tag{4}$$

which is one of the polynomial conditions for conic correspondence of $C$ and $C'$. It could be used to match conics between images by computing its value and matching conics with the lower score.

## 4.3 Closed form solution of reconstruction

To compute the quadric $\mathcal{Q}$ corresponding to the 2 planes we compute first $\lambda$ as:

$$\lambda = -\frac{I_3}{2I_2}$$

$$\text{then } \mathcal{Q} = \mathbf{Q} + \lambda\mathbf{Q}'$$

### 4.3.1 Planes extraction

Using $a_1(\lambda) = 0$ and $a_0(\lambda) = 0$ we can rewrite the polynomial characteristic (3):

$$|\mathcal{Q}(\lambda) - \mu I| = \mu^2\left(\mu^2 + a_3(\lambda)\mu + a_2(\lambda)\right) = 0$$

We obtain the two nonzero eigenvalues as roots of:

$$\mu^2 + a_3(\lambda)\mu + a_2(\lambda) = 0$$

As $\mathcal{Q}$ is a real symmetric matrix, there exists a non singular transformation $T$ such that:

$$\text{diag}(\mu_1, \mu_2, 0, 0) = \mathbf{T}^t\mathcal{Q}\mathbf{T}$$

The quadric defined by $\mathbf{x}^t\mathcal{Q}\mathbf{x} = 0$ is given in the new frame by $\mathbf{x} = \mathbf{T}\mathbf{x}'$:

$$\mu_1 x'^2 + \mu_2 y'^2 = 0$$

This could be changed to:

$$\left(\sqrt{\mu_1}x' + \sqrt{-\mu_2}y'\right)\left(\sqrt{\mu_1}x' - \sqrt{-\mu_2}y'\right) = 0$$

This gives us the equations of the two planes:

$$\left(\sqrt{\mu_1}x' \pm \sqrt{-\mu_2}y'\right) = 0$$

These equations have real solutions only if $\mu_1\mu_2 < 0$. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the eigenvectors corresponding to the eigenvalues $\mu_1$ and $\mu_2$, we obtain the plane equations in the world frame coordinates as:

$$\mathbf{p}_{1,2} \equiv \left(\sqrt{\mu_1}\mathbf{v}_1 \pm \sqrt{\mu_2}\mathbf{v}_2\right)^t \mathbf{x} = 0$$

The conic equation is given by the intersection of one of these planes with one of the projection cones.

### 4.3.2 Plane selection

The ambiguity between the two solutions can be removed in some special cases. If we restrict to Euclidean reconstruction and assume that the conic is non-transparent, i.e. is visible from two different viewpoints if only these viewpoints are located on the same side of the conic plane (Fig. 6). We can check the position of the viewpoint with respect to a conic plane given the projection centers of the two views $\mathbf{o} = \ker(\mathbf{P})$ and $\mathbf{o}' = \ker(\mathbf{P}')$. If $(\mathbf{o}^t\mathbf{p}_i)\,(\mathbf{o}'^t\mathbf{p}_i) > 0$ is satisfied, both $\mathbf{o}$ and $\mathbf{o}'$ lie on the same side of the plane $\mathbf{p}_i$.

In the projective case, to use this predicate, we must assume that the conic has no point at infinity [8] (i.e. only ellipses).

### 4.3.3 Conic extraction

Given the previous results we are able to reconstruct the plane in which the conic lies. We now compute the intersection of this plane $\mathbf{p}$ with one projective cone, and deduce its equation.

We choose a new frame $\mathbf{L}$ so that $\mathbf{p} \equiv z' = 0$ in this frame. We call $\mathbf{R}$ the transformation such that $\mathbf{x} = \mathbf{R}\mathbf{x}'$. The cone $\mathrm{Q} \equiv \mathbf{x}^t\mathbf{Q}\mathbf{x} = 0$ in the new frame is expressed as $\mathrm{Q}' \equiv \mathbf{x}'^t\mathbf{Q}'\mathbf{x}' = 0$, where $\mathbf{Q}' = \mathbf{R}^t\mathbf{Q}\mathbf{R}$. As we constrain the conic plane to be $z' = 0$ we get:

$$\mathrm{Q}' \equiv \mathbf{x}'^t\mathbf{Q}'\mathbf{x}' = 0 \Leftrightarrow (tx', ty', tz', t)\begin{pmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & i \\ d & g & i & j \end{pmatrix}\begin{pmatrix} tx' \\ ty' \\ tz' \\ t \end{pmatrix} = 0$$

$$\Rightarrow (tx', ty', 0, t)\begin{pmatrix} a & b & - & d \\ b & e & - & g \\ - & - & - & - \\ d & g & - & j \end{pmatrix}\begin{pmatrix} tx' \\ ty' \\ 0 \\ t \end{pmatrix} = 0$$

$$\Rightarrow \mathcal{C}' \equiv (tx', ty', t)\,\mathcal{C}'\begin{pmatrix} tx' \\ ty' \\ t \end{pmatrix} = 0 \text{ with } \mathcal{C}' = \begin{pmatrix} a & b & d \\ b & e & g \\ d & g & j \end{pmatrix}$$

$\mathcal{C}'$ is the equation of $\mathcal{C}$ in the plane frame. We can transform it again to express it in its reference frame as $\mathcal{C}$.

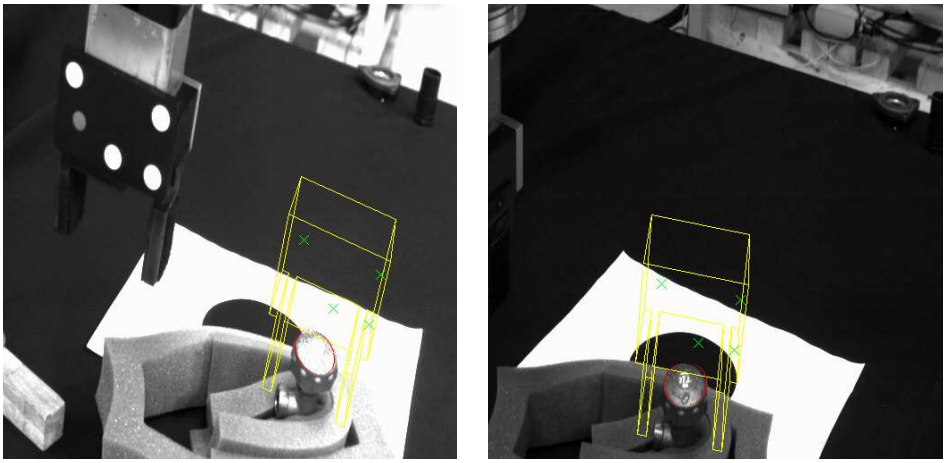

Figure 7: A typical stereo image pair showing fitted ellipses (in red) and predicted gripper positions for optimal grasping.

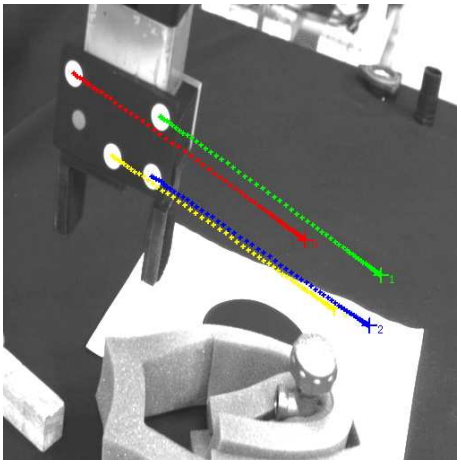

Figure 8: Gripper tracking: each point represents a gripper mark during tracking.

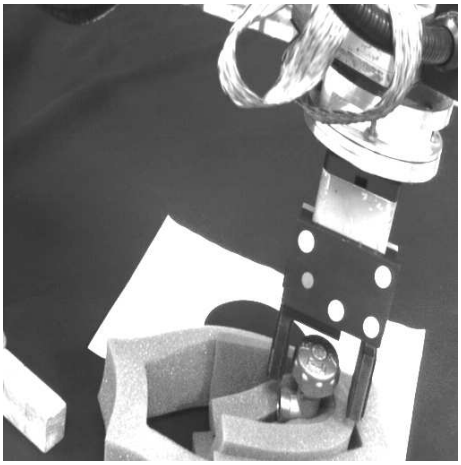

Figure 9: Final gripper position.

## 5 Experiment and Conclusion

This work is a continuation of [8] and [1]. Experiments to study robustness of the pose algorithm have been performed in [8] (mostly by simulation). Real experiments have been performed using a circular calibration pattern. They show that on average, the circle is reconstructed as an ellipse with an axis ratio of 0.99. There is one degenerated case, when the elliptical plane is parallel to the plane defined by the optical axis of the stereo pair. Experiments of the servoing algorithm itself is done in [1] and [2], which study its sensibility to intrinsic parameters.

Using the method just described, we have experimentaly aligned a gripper with a gas valve located approximately 1.5 meters away from the stereo rig. This alignment was correct within 1mm in position – considering the distance between the gripper and the valve axes – and 0.5 degrees in orientation (Fig. 7, 8,and 9).

The overall method itself seems very robust. Indeed, the system was calibrated once, and experiments were run over the following weeks, without any re-calibration. Hence this method seems well suited for practical applications in the robotic and telemanipulation fields. Future work includes studying in a mathematically more strict way, why the experiments with this method work well, especially the pose algorithm, even when the intrinsic parameters are known without high accuracy.

# References

[1] F. Dornaika. *Contributions à l'intégration vision/robotique : calibrage, localisation et asservissement*. Thèse de doctorat, Institut National Polytechnique de Grenoble, LIFIA–IMAG–INRIA Rhône-Alpes, September 1995.

[2] B. Espiau. Effect of camera calibration errors on visual servoing in robotics. In *Proceedings of the Third International Symposium on Experimental Robotics, Kyoto, Japan*, pages 187–193, October 1993.

[3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. IEEE *Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[4] G.D. Hager, W. Chang, and A.S. Morse. Robot hand-eye coordination based on stereo vision. IEEE *Control Systems*, page 30, 1995.

[5] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 579–587. Springer-Verlag, 1992.

[6] N. Hollinghurst and R. Cipolla. Uncalibrated stereo hand-eye coordination. In J. Illingworth, editor, *Proceedings of the fourth British Machine Vision Conference, Surrey, England*, volume 2, pages 389–398. British Machine Vision Association, BMVA Press, September 1993.

[7] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose: The link between weak perspective, paraperspective and full perspective. *International Journal of Computer Vision*, 22(2):173–189, March 1997.

[8] L. Quan. Conic reconstruction and correspondence from two views. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 18(2):151–160, February 1996.

[9] J.G. Semple and G.T. Kneebone. *Algebraic Projective Geometry*. Oxford Science Publication, 1952.