

## Visual Servoing Based on Dynamic Vision

Ali Alhaj

Christophe Collewet

François Chaumette

*Cemagref*

17 Avenue de Cucillé  
35044 Rennes Cedex, France  
ali.alhaj@cemagref.fr

*Cemagref*

17 Avenue de Cucillé  
35044 Rennes Cedex, France  
christophe.collewet@cemagref.fr

IRISA / INRIA Rennes

Campus Universitaire de Beaulieu  
35042 Rennes Cedex, France  
francois.chaumette@irisa.fr

### Abstract

*This paper deals with the way to achieve positioning tasks by visual servoing in the case of planar and motionless objects whose shape is unknown. In fact, we consider here complex objects like those one can encounter in the agrifood industry. A 3D reconstruction phase is first considered yielding to the parameters of the plane. This computation is based on the measurement of the 2D velocity in a region of interest and on the measurement of the camera velocity. Once the parameters of the plane are sufficiently stable, a visual servoing scheme is used to control the orientation of the camera with respect to the object and to ensure that it remains in the camera field of view for any desired orientation. This 3D reconstruction phase is maintained active during the servoing to improve the accuracy of the parameters and, consequently, to obtain a small positioning error. Finally, experimental results validate the proposed approach as well as for its effectiveness than for the obtained accuracy.*

### 1 Introduction

2D visual servoing allows the realization of robotic tasks directly from visual features acquired by a camera. These features are compared with the desired ones, extracted from the desired position of the camera with respect to the considered object[1]. Nevertheless, we still cannot achieve positioning tasks with regard to partially known objects. Indeed, except rigid manufactured goods for which a model often exists, we rarely have a precise description of the object or of the desired visual features, either because these objects can be subject to deformations or simply because of their natural variability. Such cases can appear for example in surgical domain, agrifood industry, agriculture or in unknown environments (underwater, space). In [2], the authors use a specific motion to perform

an alignment task without a precise description of the desired visual features. Unfortunately, their study is restricted to planar motions. In [3], thanks to dynamic visual features, a positioning task consisting in moving the camera to a position parallel to a planar object of unknown shape is achieved. However, such an approach needs particular motion parameters estimation [4] currently leading to a high computation duration and, consequently, to a low control scheme rate. In addition, this approach cannot be used for any specified orientation of the camera. This case has been taken into account in [5], where geometric features are used. This approach is based on the maximization in the image of the surface of a triangle built from three feature points. To do that, three tasks have to be performed sequentially yielding, in some cases, to excessive durations of the task. Moreover, tracking the feature points can be difficult, depending, in the case of agrifood products, on their texture.

The approach described in this paper proposes to treat the same problem, that is the realization of positioning tasks with respect to a planar and motionless object of unknown shape for any specified orientation of the camera. Since the shape of the object is considered as unknown, a 3D reconstruction phase by *dynamic vision* is first performed. This computation is based, as in [3], on the measurement of the 2D motion in a region of interest, but here we use it to recover the structure of the object. This way to proceed allows, as will be shown later, more flexibility to synthesize the control law, in particular to ensure that the object remains in the camera field of view.

The paper is organized as follows: first, we present in Section 2 a brief review on previous works relevant to 3D reconstruction by dynamic vision. We formulate then the problem in Section 3 and describe how to obtain the structure of the object in Section 4. Section 5 details the way we synthesize the control law. Finally, experimental results concerning objects of unknown shape are presented in Section 6.

## 2 Previous works

Let us consider a point  $P$  of the object described by  $\underline{P} = (X, Y, Z)^T$  in the camera frame, with the  $Z$  axis the camera optical axis. Assuming without loss of generality a unit focal length, this point projects in  $p$ , described by  $\underline{p} = (x, y, 1)^T$ , according to

$$\underline{p} = \frac{\underline{P}}{Z} \quad (1)$$

which yields to the well-known relation [6]

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -1-x^2 & y \\ 0 & 1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} T_c \quad (2)$$

where  $T_c = (\underline{V}^T, \underline{\Omega}^T)^T$  is the camera velocity and  $\underline{V} = (V_x, V_y, V_z)^T$  and  $\underline{\Omega} = (\Omega_x, \Omega_y, \Omega_z)^T$  its translational and rotational components respectively. This relation can be rewritten as follows

$$\begin{cases} \dot{x} = \frac{-V_x + xV_z}{Z} + xy\Omega_x - (1+x^2)\Omega_y + y\Omega_z \\ \dot{y} = \frac{-V_y + yV_z}{Z} + (1+y^2)\Omega_x - xy\Omega_y - x\Omega_z. \end{cases} \quad (3)$$

In this equation, only the depth  $Z$  is unknown if  $\underline{p}$ ,  $\dot{\underline{p}}$  and  $T_c$  can be measured.

Various ways to estimate  $Z$  exist, they are based on different approaches to cope with  $\dot{\underline{p}}$ . The most immediate way is to approximate the velocities  $\dot{x}$  ( $\dot{y}$ ) by  $\frac{\Delta x}{\Delta t}$  ( $\frac{\Delta y}{\Delta t}$ ) [7]. However, this method does not provide accurate results because of errors introduced by the discretization. A way to greatly improve these results is to impose  $\dot{x} = \dot{y} = 0$  [8]. This may be realized easily using visual servoing, this approach is then relevant to *active vision*. Another approach is based on the assumption that the brightness of  $p$  remains constant during the motion. This assumption leads to the well-known additional constraint [6]

$$\dot{x}I_x + \dot{y}I_y + I_t = 0 \quad (4)$$

where  $I_x$ ,  $I_y$  and  $I_t$  represent the spatio-temporal derivatives of the intensity of the considered point in the image. By substituting  $\dot{x}$  and  $\dot{y}$  given by (3) in (4), an expression of  $Z$  can be obtained [9, 10] (note that these works treat the more general case where  $T_c$  is also supposed to be unknown). Such approaches, known as *direct approaches*, require accurate estimations of the derivatives  $I_x$ ,  $I_y$  and  $I_t$  and therefore, are not very accurate in practice. Another way is to locally model the surface of the object in the neighborhood of  $P$ . That provides an expression of  $1/Z$  in function of the chosen parameterization, which can be used in (3) to exhibit a parametric model of the 2D

motion. On the other hand, these parameters can be obtained by a method of computation of the 2D motion, like in [4] for example. Finally, an expression of the structure of the object can be extracted [11] (here too, by considering a second point, the case where  $T_c$  is unknown is treated). These approaches are known as *indirect approaches* since they require an intermediate computation of the 2D motion.

The main benefit of our approach with regard to the previous works is that we explicitly use parameters obtained by 3D reconstruction in the control scheme. This allows us to compute the orientation error and to synthesize easily the control law, in particular to take into account any desired orientation of the camera.

## 3 Modeling

If we assume that the considered object is planar, or at least planar in a neighborhood of  $P$ , we have

$$Z = AX + BY + C \quad (5)$$

which, according to (1), can be rewritten in function of  $\underline{p}$  as follows

$$\frac{1}{Z} = \alpha x + \beta y + \gamma \quad (6)$$

with  $\alpha = -\frac{A}{C}$ ,  $\beta = -\frac{B}{C}$  and  $\gamma = \frac{1}{C}$ . By substituting this expression in (3), one can show that the 2D motion is exactly modeled by a parametric model with 8 parameters [12]. By neglecting the second order terms, we obtain

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} a_0 + a_1x + a_2y \\ b_0 + b_1x + b_2y \end{pmatrix} \quad (7)$$

where

$$\begin{cases} a_0 = -\Omega_y - \gamma V_x \\ a_1 = -\alpha V_x + \gamma V_z \\ a_2 = \Omega_z - \beta V_x \\ b_0 = \Omega_x - \gamma V_y \\ b_1 = -\Omega_z - \alpha V_y \\ b_2 = -\beta V_y + \gamma V_z \end{cases} \quad (8)$$

which can be rewritten under a matrix form as follows

$$M\underline{\Theta} = \underline{\Gamma} \quad (9)$$

with

$$M = \begin{pmatrix} 0 & 0 & -V_x \\ -V_x & 0 & V_z \\ 0 & -V_x & 0 \\ 0 & 0 & -V_y \\ -V_y & 0 & 0 \\ 0 & -V_y & V_z \end{pmatrix}, \underline{\Gamma} = \begin{pmatrix} a_0 + \Omega_y \\ a_1 \\ a_2 - \Omega_z \\ b_0 - \Omega_x \\ b_1 + \Omega_z \\ b_2 \end{pmatrix} \quad (10)$$

and

$$\underline{\Theta} = (\alpha \quad \beta \quad \gamma)^T. \quad (11)$$

Using a measure of the 2D motion parameters and the camera velocity,  $\underline{\Theta}$  can be easily obtained by solving (9):

$$\underline{\Theta} = (M^T M)^{-1} M^T \underline{\Gamma}. \quad (12)$$

However, the solution is correct only if the matrix  $M^T M$  is well conditioned, that is if

$$\min(\nu_i) > \nu_s, \quad i = 1 \dots 3 \quad (13)$$

where the  $\nu_i$ 's are the eigenvalues of  $M^T M$  and  $\nu_s$  a given threshold. We will use this property in Section 6.

Next Sections 4 and 5 show respectively how to compute the 2D motion parameters and to synthesize the control law to achieve the positioning task.

#### 4 Estimation of the 2D motion parameters

The algorithms for the computation of 2D motion parameters given by (7) are relatively complex and not very easy to implement. In addition, they are expensive with regard to the computational cost. In fact, it is more judicious to investigate an approach based on the measurement of the displacement in the image rather than on the use of the velocity. This approach is similar to track the point  $p$  from an image to another. We review here the most classical approach [13] and we will see that its degree of complexity does not exceed that of the inversion of a 6x6 matrix.

Here too, we assume that the brightness of  $p$  remains unchanged during the motion, so we can write

$$I(x, y, t) = I(x + \dot{x}\Delta t, y + \dot{y}\Delta t, t + \Delta t) \quad (14)$$

where  $\Delta t$  represents the control scheme period, and  $\dot{x}$  and  $\dot{y}$  are modeled by (7).

Because of the noise, (14) is generally not satisfied. Therefore, a solution is to move the problem to an optimization one to find the parameters which have to minimize the following residue

$$\epsilon = \sum_W (I(x, y, t) - I(x + \dot{x}\Delta t, y + \dot{y}\Delta t, t + \Delta t))^2 \quad (15)$$

where  $W$  is a window of interest centered in  $p$ .

To carry out the optimization, we have to assume that  $\Delta t$  and the displacements are sufficiently small. If so, a first order Taylor expansion of  $I(x + \dot{x}\Delta t, y + \dot{y}\Delta t, t + \Delta t)$  can be performed and substituted in (15) to obtain a linear system in function of the required

parameters. Usually, this system is inverted by using an iterative Newton-Raphson style algorithm to take into account the error introduced by the Taylor expansion.

To ensure the convergence of the minimization process,  $p$  is selected from points of interest extracted from the first image. Moreover, we choose  $p$  as the best point in the sense that it will be correctly tracked during the motion. To do that, the method described in [14] has been used.

#### 5 Control law

First, let us remember the task to achieve. The goal is to ensure a given final orientation of the camera with respect to plane  $\pi$  described by (5) and, also to ensure that  $P$  will still remain in the camera field of view.

Once  $\underline{\Theta}$  is estimated, the unit normal  $\underline{n}$  of plane  $\pi$  in  $P$  in the camera frame can be derived. However, in the case of any orientation we rather have to consider  $\underline{n}^* = \mathcal{R}\underline{n}$  where  $\mathcal{R}$  is the rotation matrix computed from the desired orientation (see Figure 1). Therefore, we have to move the camera so that  $\underline{Z} = \underline{n}_c$  with  $\underline{Z}$  the unit vector carried by the optical axis and  $\underline{n}_c = -\underline{n}^*$ . This rotation to perform can be expressed under the form  $\underline{u}\theta$  where  $\underline{u}$  represents the rotation axis and  $\theta$  the rotation angle around this axis

$$\underline{u} = \frac{\underline{Z} \wedge \underline{n}_c}{\|\underline{Z} \wedge \underline{n}_c\|} \quad (16)$$

and

$$\theta = \arccos(\underline{Z} \cdot \underline{n}_c) \quad (17)$$

The camera orientation being known, it is possible to compute the control law. We used the one described in [15]. Indeed, it ensures that  $P$  remains in the camera field of view since the trajectory of  $p$  is a straight line between the current position  $p$  and the desired position  $p^*$  (which has been chosen as the principal point of the image). We describe here briefly this approach known as *hybrid visual servoing*.

First,  $\underline{p}_r$  is defined as follows

$$\underline{p}_r = \frac{1}{Z^*} \underline{P} = \frac{Z}{Z^*} \underline{p} \quad (18)$$

with  $Z^*$  the desired depth for  $P$  in final position. One can then show that

$$\dot{\underline{p}}_r = \left( -\frac{1}{Z^*} \mathbb{I}_3 [\underline{p}_r]_{\times} \right) T_c \quad (19)$$

where the notation  $[\underline{v}]_{\times}$  denotes the antisymmetric matrix associated to  $\underline{v}$ .

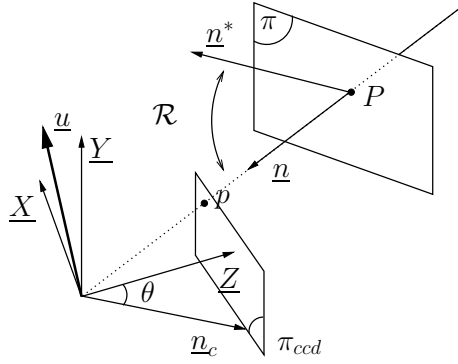


Figure 1: Rotation to perform by the camera.

The change of the orientation can be expressed as

$$\frac{d(\underline{u}\theta)}{dt} = \begin{pmatrix} 0 & L_\Omega \end{pmatrix} T_c \quad (20)$$

with

$$L_\Omega(\underline{u}, \theta) = \mathbb{I}_3 - \frac{\theta}{2} [\underline{u}]_\times + \left( 1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2\left(\frac{\theta}{2}\right)} \right) [\underline{u}]_\times^2 \quad (21)$$

where  $\text{sinc}(\theta) = \sin(\theta)/\theta$ .

Expressions (19) and (20) can be merged as follows

$$\frac{d}{dt} \begin{pmatrix} \underline{p}_r \\ \underline{u}\theta \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z^*} \mathbb{I}_3 & [\underline{p}_r]_\times \\ 0 & L_\Omega \end{pmatrix} T_c = LT_c \quad (22)$$

Finally, the desired positioning task can be expressed in term of regulation to zero of the following task function

$$\underline{\epsilon} = \begin{pmatrix} \underline{p}_r - \underline{p}_r^* \\ \underline{u}\theta \end{pmatrix} \quad (23)$$

yielding to the camera velocity

$$T_c = -\lambda \widehat{L}^{-1} \underline{\epsilon} \quad (24)$$

$\lambda$  being a positive gain and  $\widehat{L}$  an approximation of  $L$ .

Let us note that the values of  $Z$  and  $\underline{p}$  required for the computation of  $\underline{p}_r$  are obtained respectively thanks to (6) and by integration of (7).

## 6 Experimental results

In order to validate the proposed algorithm, we present here experimental results for three different desired orientations. The experimental system is described in [5].

The object consists of a photograph of a trout steak fixed on a planar support. To evaluate the positioning

accuracy of our method, this support makes possible to express precisely the transformation matrix between the camera frame and the object one with the method used in [5]. This matrix is characterized by the Euler's angles denoted  $\phi_X, \phi_Y, \phi_Z$  which respectively represent the angles of the  $X, Y$  and  $Z$  rotations.

Furthermore, since the object is motionless, one can improve the accuracy on  $\underline{\Theta}$ . Indeed, in a fixed frame, one can express a value  $\underline{\Theta}^f$  that can be filtered since a fixed value has to be obtained. Thereafter, this value is expressed in the camera frame to be used in the control law. Moreover, proceeding this way allows to know when  $\underline{\Theta}^f$  is stable enough to be used in the control law (typically fifteen acquisitions are sufficient). Thus, a preliminary phase with a constant velocity is required. We imposed  $V_x = V_y = 2$  cm/s,  $V_z = 0$  and  $\underline{\Omega} = 0$ .

Finally, the algorithm consists of three phases, a first phase at constant velocity, a second phase when both reconstruction and servoing are performed, and a last phase where only the servoing operates. This last phase occurs when the constraint given by (13) is no more satisfied (which occurs near convergence when the motion is very small).

The following values were used for all the experiments:  $\lambda = 0.06$ ,  $\nu_s = 1.10^{-6}$ ,  $\Delta t = 800$  ms,  $Z^* = 70$  cm and the size of  $W$  has been fixed to  $55 \times 55$  pixels.

The first experiment consists in positioning the camera parallel to  $\pi$ . Figures 2.a depicts respectively the behavior of parameters  $A, B$  and  $C$  in a fixed frame; Figures 2.b the components of  $\underline{n}$ ; Figures 2.c-d the behavior of the motion parameters, respectively  $a_0, b_0$  and  $a_i, b_i$  ( $i = 1, 2$ ). The components of the camera velocity are represented on Figure 2.e, the normalized error on Figure 2.f (defined as  $\frac{\|\underline{\epsilon}\|_{t=k\Delta t}}{\|\underline{\epsilon}\|_{t=0}}$ ), the estimated depth on Figure 2.g and finally, the initial and final images are reported respectively on Figures 2.h-i.

First, Figure 2.f confirms that the control law converges since the normalized error tends towards zero, as well as  $\widehat{Z}$  towards  $Z^*$  (Figure 2.g). In the same way, as expected,  $p$  tends towards the principal point following a straight line (see Figure 2.i where the trajectory of  $p$  is drawn, the first segment corresponding to the motion relative to the preliminary phase). One can also remark on Figure 2.a that the reconstruction stops around the 120<sup>th</sup> iteration, while the stop condition relative to the normalized error is not reached before the 300<sup>th</sup> iteration approximately. For this experiment, the initial orientation of the camera was  $\phi_X = 9.66^\circ$ ,  $\phi_Y = 23.77^\circ$  and  $\phi_Z = 9.46^\circ$ , the orientation after the servoing was  $\phi_X = 2.44^\circ$  and  $\phi_Y = 1.39^\circ$  (let us recall that  $\phi_Z$  is not controlled).

Consequently, the positioning error is less than  $2.5^\circ$ .

The second experiment consists in positioning the camera so that  $\phi_X = 0^\circ$  and  $\phi_Y = 20^\circ$ . Figures 3.a-i describe the behavior of the same variables as for the previous experiment. The same comments can be made, in particular concerning the convergence of the control law. However, as seen in Figure 3.i, the trajectory of  $p$  does not follow a straight line, probably, because of a relatively bad behavior of the point tracking algorithm. But this phenomenon has no bad consequence, since the positioning error remains small: the final orientation is  $\phi_X = -0.65^\circ$  and  $\phi_Y = 20.16^\circ$ . For this experiment the initial orientation was  $\phi_X = 16.88^\circ$ ,  $\phi_Y = 11.22^\circ$  and  $\phi_Z = 7.40^\circ$ .

For the last experiment the desired orientation was  $\phi_X = -10^\circ$  and  $\phi_Y = 0^\circ$ . Figures 4.a-i describe the same variables as previously. Here again, the positioning error is small, we have  $\phi_X = -8.90^\circ$  and  $\phi_Y = -0.06^\circ$ . The initial orientation was  $\phi_X = 6.58^\circ$ ,  $\phi_Y = -10.94^\circ$  and  $\phi_Z = 2.83^\circ$ .

## 7 Conclusion and future works

We have presented in this paper a way to achieve positioning tasks by visual servoing when the desired image of the object cannot be precisely described and for any desired orientation of the camera. However, we have to assume the object to be planar and motionless. The approach is based on a 3D reconstruction which allows the estimation of the current orientation of the object with respect to the camera, and thereafter to the elaboration of the control law. Experimental results validated our algorithm, low positioning errors were observed ( $\approx 2^\circ$ ). However, we can regret that our method is quite sensitive to the calibration of the robot since it necessitates a measurement of the camera velocity.

In the future, an interesting prospect is to extend the algorithm to the case of nonplanar objects. The algorithm should not be too much modified by considering locally a parametric modeling around  $P$ .

## References

- [1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.
- [2] B. Yoshimi and P. K. Allen, "Active uncalibrated visual servoing," in *IEEE Int. Conf. on Robotics and Automation, ICRA '94*, San Diego, May 1994, pp. 156–161.

- [3] A. Crétual and F. Chaumette, "Visual servoing based on image motion," *Int. Journal of Robotics Research*, vol. 20, no. 11, pp. 857–877, November 2001.
- [4] J.M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, December 1995.
- [5] C. Collewet and F. Chaumette, "Positioning a camera with respect to planar objects of unknown shape by coupling 2-d visual servoing and 3-d estimations," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 3, pp. 322–333, June 2002.
- [6] B.K.P. Horn and B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 16, no. 1–3, pp. 185–203, August 1981.
- [7] P. Rives and M. Xie, "Toward dynamic vision," in *Proc. IEEE Workshop on Interpretation of 3D scenes*, Austin, Texas, November 1989.
- [8] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 492–504, May 1996.
- [9] B. K. P. Horn and E. J. Weldon, "Direct methods for recovering motion," *Int. Journal of Computer Vision*, vol. 2, no. 1, pp. 51–76, June 1988.
- [10] S. Negahdaripour and B. K. P. Horn, "Direct passive navigation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 168–176, January 1987.
- [11] S. Negahdaripour and S. Lee, "Motion recovery from image sequences using only first order optical flow information," *Int. Journal of Computer Vision*, vol. 9, no. 3, pp. 163–184, 1992.
- [12] G. Adiv, "Determining 3d motion and structure from optical flow generated by several moving objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384–401, July 1985.
- [13] J. Shi and C. Tomasi, "Good features to track," in *IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, Seattle, June 1994, pp. 593–600.
- [14] C. Kermad and C. Collewet, "Improving feature tracking by robust points of interest selection," in *6th International Fall Workshop on Vision, Modeling, and Visualization, VMV'2001*, Stuttgart, Germany, November 21–23, 2001, pp. 415–422.
- [15] E. Malis and F. Chaumette, "Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 176–186, April 2002.

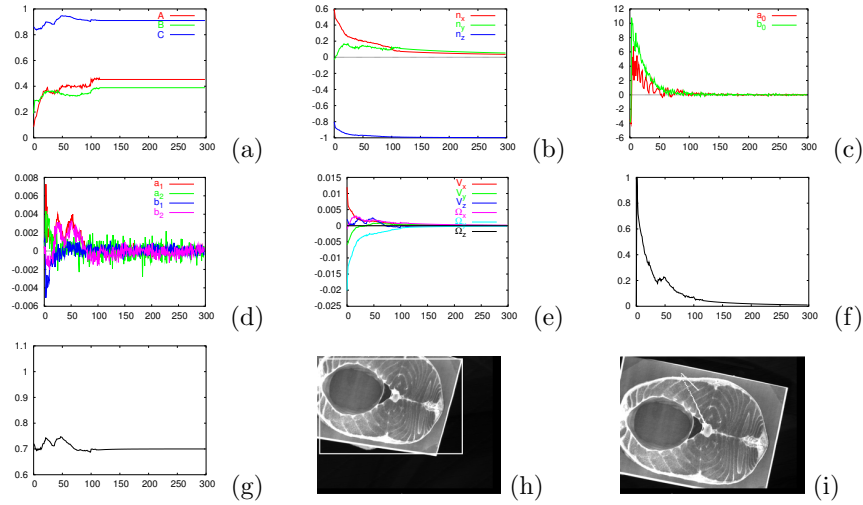


Figure 2: 1<sup>st</sup> experiment: (a) Parameters of the plane in a fixed frame. (b) Components of the normal  $\underline{n}$  in the camera frame. (c) Parameters  $a_0, b_0$ . (d) Parameters  $a_1, b_1, a_2, b_2$ . (e) Kinematic screw. (f) Normalized error. (g) Estimated depth  $\hat{Z}$ . (h) Initial image. (i) Final image.

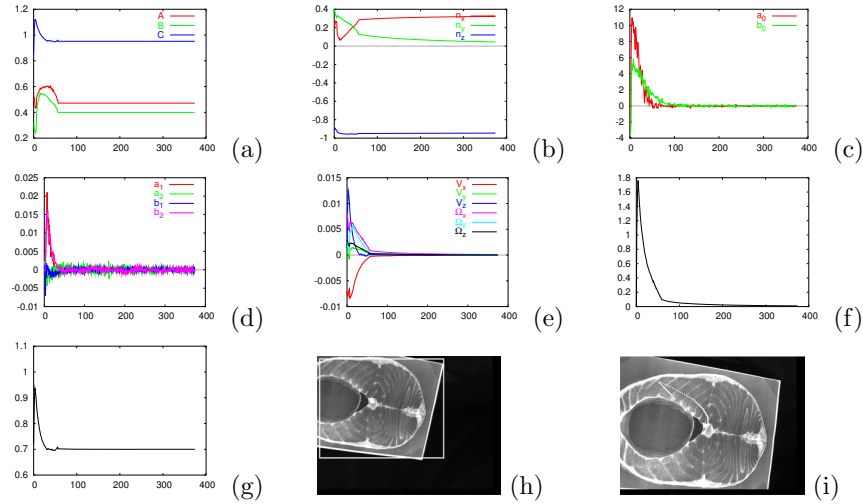


Figure 3: 2<sup>nd</sup> experiment.

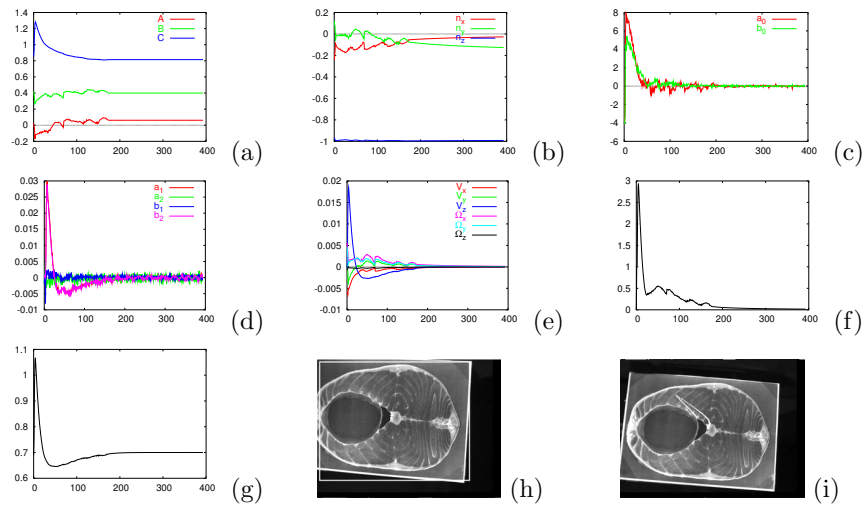


Figure 4: 3<sup>rd</sup> experiment.