HAL
open science

A camera-projector system for robot positioning by visual servoing

J. Pagès, Christophe Collewet, François Chaumette, J. Salvi

To cite this version:

J. Pagès, Christophe Collewet, François Chaumette, J. Salvi. A camera-projector system for robot positioning by visual servoing. IEEE Int. Workshop on Projector-Camera Systems, PROCAM'2006, 2006, New York, France. pp.2-9. inria-00350319

HAL Id: inria-00350319

https://inria.hal.science/inria-00350319v1

Submitted on 6 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A camera-projector system for robot positioning by visual servoing

Jordi Pagès
Institut d'Informàtica i Aplicacions
University of Girona
Girona, Spain
jpages@eia.udg.es

Christophe Collewet
IRISA/ INRIA Rennes
Campus de Beaulieu
Rennes, France
christophe.collewet@irisa.fr

François Chaumette
IRISA/ INRIA Rennes
francois.chaumette@irisa.fr

Joaquim Salvi
Institut d'Informàtica i Aplicacions
qsalvi@eia.udg.es

Abstract

Positioning a robot with respect to objects by using data provided by a camera is a well known technique called visual servoing. In order to perform a task, the object must exhibit visual features which can be extracted from different points of view. Then, visual servoing is object-dependent as it depends on the object appearance. Therefore, performing the positioning task is not possible in presence of non-textured objets or objets for which extracting visual features is too complex or too costly. This paper proposes a solution to tackle this limitation inherent to the current visual servoing techniques. Our proposal is based on the coded structured light approach as a reliable and fast way to solve the correspondence problem. In this case, a coded light pattern is projected providing robust visual features independently of the object appearance.

1. Introduction

Visual servoing is a largely used technique which is able to control robots by using data provided by visual sensors. The most typical configuration is eye-in-hand, which consists of linking a camera to the end-effector of the robot. Then, typical task of positioning the robot with respect to objects or target tracking are fulfilled by using a control loop based on visual features extracted from the images [8].

All the visual servoing techniques assume that it is possible to extract visual measures from the object in order to perform a pose or partial pose estimation or to use a given set of features in the control loop. Therefore, visual servoing does not bring any solution for positioning with respect to non-textured objects or objects for which extracting visual features is too complex or too time consuming. Note that the sampling rate in visual servoing must be high enough in order to not penalising the dynamics of the end-effector and the stability of the control scheme.

A possible solution to this problem is to project structured light on the objects in order to obtain visual features. There are few works in this field and they are mainly based on the use of laser pointers and laser planes [1, 10, 15]. Furthermore, they are usually designed for positioning with respect to planar objects [14] or specific non-planar objects like spheres [10]. In this paper, we propose the use of coded structured light [2]. This is a powerful technique based on the projection of coded light patterns which provide robust visual features. This technique has been largely used in shape acquisition applications based on triangulation. However, it has never been used in a visual servoing framework. With the use of coded patterns, visual features are available independently of the object appearance so that visual servoing techniques can tackle their limitation in front of non-textured objects. However, in case of moving objects it has several problems as the projected features do not remain static on the object surface. In a first attempt to combine coded structured light with visual servoing this paper only considers static objects. The paper is structured as follows. In Section 2 the coded light approach and its ability to provide visual features is presented. Then, the coded pattern used in this work is presented in Section 3. Afterwards, Section 4 briefly reviews the formalism of a positioning task by using visual servoing and the control law based on the visual features provided by the coded pattern. Experiments validating the approach are shown in Section 5. Section 6 points out some advantages and limitations of the approach. Finally, the end of the paper discusses conclusions and future works.

2. Providing visual features with a coded pattern

Coded structured light is considered as an active stereovision technique [9]. It is said to be active because controlled illumination is used in order to simplify computer vision tasks. The typical configuration consists of a DLP (Digital Light Projector) projector and one (or two) camera(s). In both cases the DLP projector is used for projecting a light pattern on the object. The advantage of using a DLP projector is that patterns of high resolution and large number of colours can be projected. Furthermore, a high flexibility is obtained as the projected pattern can be changed with no cost. In case of using a unique camera, the projector is considered as an inverse camera and correspondences between the perceived image and the projected pattern are easily found. The effectiveness of coded structured light rely on the coding strategy used for defining the patterns. Typically, codification allows a set of pattern points or lines to be uniquely identified. Then, the decoding process consists in locating encoded points or lines in the image provided by the camera when the pattern is being projected on the object. The typical application of coded structured light is shape acquisition [2]. In this case, the correspondences are triangulated obtaining a 3D reconstruction of the object view. This is possible if the camera and the projector have been accurately calibrated previously. Nevertheless, the aim of coded structured light is to provide robust, unambiguous and fast correspondences between the projected pattern and the image view.

A large number of different coded structured light techniques exist [2]. Among all the techniques, there are two main groups: time-multiplexing and one-shot techniques. Time-multiplexing techniques are based on projecting a sequence of binary or grey-scaled patterns. The advantage of these techniques is that, as the number of patterns is not restricted, a large resolution, i.e. number of correspondences, can be achieved. Furthermore, binary patterns are robust against the object's colour. On the other hand, their main constraint is that during the projection of the patterns the object, the projector and the camera must all remain static (which is incompatible in visual servoing). One-shot techniques project a unique pattern so that a moving camera or projector can be considered. In order to concentrate the codification scheme in a unique pattern, each encoded point or line is uniquely identified by a local neighbourhood around it. Then, for correctly decoding the pattern in the image, the object surface is assumed to be locally smooth. Otherwise, every encoded neighbourhood can appear incomplete in the image which can provoke decoding errors.

From the point of view of visual servoing, one-shot coded structured light is a powerful solution for robustly providing correspondences independently of the object appearance. By projecting a coded pattern, correspondences are easily found between the reference image and the initial and intermediate images. In Fig. 1 several one-shot patterns projected on a horse statue are shown.

In the first pattern, every coloured spot is uniquely encoded by the window of 3×3 spots centred on it. In the second pattern, every stripe is uniquely encoded by its colour and the colour of the adjacent stripes. Finally, the third pattern uses the same codification for both horizontal and vertical slits.

The choice of the coded pattern for visual servoing depends on the object, the number of correspondences that we want to get, the lighting conditions, the required decoding time, etc. In this paper, the pattern chosen is a coloured array of spots like the one shown in Fig. 1a. The main reason is that it can be decoded very fast, according to the control rate requirements. Furthermore, it provides image point correspondences which are useful for many of the existing visual servoing techniques.

In the following section the pattern used in the experiments is presented in more details.

3. Pattern based on M-array codification

Many patterns encoding points have been proposed in the literature [4, 7, 13, 18]. A comprehensive state-of-art has been recently published in [16]. Most of the point-based patterns use the theory of pseudo-random arrays, also known as *M-arrays*, for encoding the points. The main advantage of such coding scheme is that high redundancy is included, increasing the robustness when decoding the pattern. Firstly, the formal definition of an M-array is briefly reviewed. Afterwards, the pattern design chosen for our application is presented. Then, an overview of the pattern decoding procedure is introduced.

3.1. Formal definition

Let $\mathbf{M}$ be a matrix of dimensions $r \times v$ where each element is taken from an alphabet of k elements $\{0, 1, .., k-1\}$. If $\mathbf{M}$ has the window property, i.e. each different submatrix of $\mathbf{M}$ of dimensions $n \times m$ appears exactly once, then $\mathbf{M}$ is a *perfect map*. If $\mathbf{M}$ contains all submatrices of $n \times m$ except the one filled by 0's, then $\mathbf{M}$ is called an *M-array* or *pseudo-random array* [11]. This kind of array has been widely used in pattern codification because the window property allows every different submatrix to be associated with an absolute position in the array. An example of a 4×6 binary M-array with window property 2×2 is

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{1}$$

a) b) c)

Figure 1. Several one-shot patterns. a) Array of dots. b) multi-slit pattern. c) Grid pattern.

This type of arrays can be constructed by folding a pseudo-random sequence [11] which is the unidimensional variant of an M-array. In this case, however, the length of the pseudo-random sequence, the size of the resulting M-array and the size of its window length are correlated. Therefore, a generic M-array of given dimensions with a given window property cannot always be constructed. An alternative algorithm sharing a similar constraint was proposed by Griffin et al. [7]. In order to cope with this constraint, an alternative consists in generating a *perfect submap*. This type of arrays has also the window property, but not all the possible windows are included. Morano *et al*. [13] proposed a brute force algorithm for generating perfect submaps.

3.2. Pattern design

There are several ways of designing a pattern with the aid of a M-array [4,7,13,18]. In most cases, patterns containing an array of spots are used like in [4, 13]. Every element of the alphabet is assigned to a grey level or a colour.

Four our visual servoing purposes, a 20×20 M-array based on an alphabet of 3 symbols $\{0, 1, 2\}$ and window property 3×3 has been generated according to the brute-force algorithm by Morano *et al*. [13]. The obtained pattern can be seen in Fig. 2 or 3a where blue has been matched with 0, green with 1 and red with 2.

3.3. Pattern segmentation and decoding

When the pattern is projected on an unknown object, the camera provides an image of the pattern deformed according to the object shape. Firstly, it is necessary to segment the pattern in the image, i.e. identifying which parts of the image contain the projected pattern. Such operation is referred as *pattern segmentation*. One of the classic advantages of using coded light is that the image processing is greatly simplified. Usually, with an appropriate camera aperture it is possible to perceive only the projected pattern removing the rest of the scene.

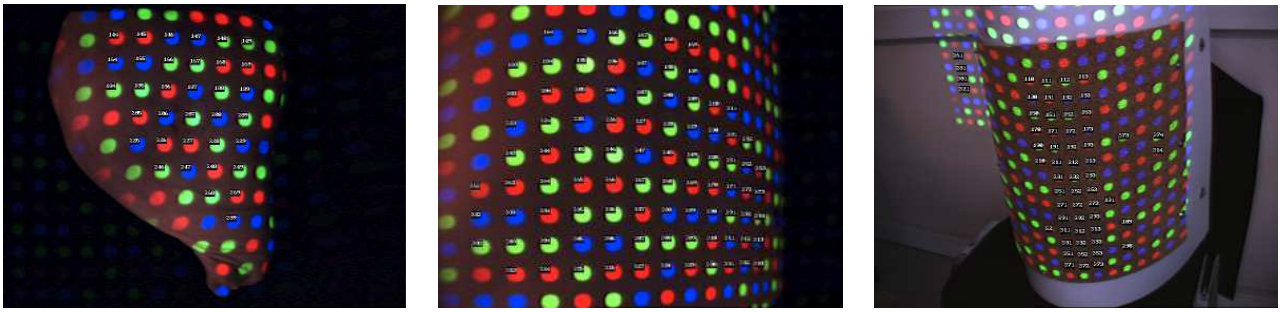
In our case, the pattern segmentation process consists in finding the visible coloured spots. Once the centre of gravity of every spot is located and its colour is identified, the decoding process can start. The steps are hereafter summarised:

- *Adjacency graph*: for every spot, the four closest spots in the four main directions are searched. With this step the 4-neighbourhood of each spot is located. Then, the 8-neighbourhood of every spot can be completed.
- *Graph consistency*: in this step, the consistency of every 8-neighbourhood is tested. For example, given a spot, its north-west neighbour must be the west neighbour of its north neighbour, and at the same time, the north neighbour of its west neighbour. These consistency rules can be extrapolated to the rest of neighbours corresponding to the corners of the 8-neighbourhood. Those spots not respecting the consistency rules are removed from the 8-neighbourhood being considered.
- *Spot decoding*: every spot having a complete 8-neighbourhood, its colour and the colours of its neighbours are used for identifying the spot in the original pattern. In order to speed up this search, a look up table storing all the 3×3 windows included in the pattern is used.
- *Decoding consistency:* every spot can be identified by the 9 windows of 3×3 in which it takes part. Those spots for which all the windows do not provide the same identification are removed.

Note that the decoding process is quite exigent and does not allow inconsistences or uncertainties. This can provoke an important number of spots to be rejected. On the other hand, this ensures a high robustness because erroneous decoded spots will rarely occur.

Examples of pattern decoding are shown in Fig. 2. The successfully decoded spots are indicated with an overprinted numeric mark. In the two first examples, the camera aperture has been adjusted in order to remove most part of the scene so that only the projected pattern is visible. In the first example, the object is a ham where most part of visible dots have been decoded. The other two examples show an object similar to an elliptic cylinder in different contexts. In Fig. 2b, the pattern is clearly visible and most part of the points are identified. On the other hand, the scene and the image shown in Fig. 2c are pretty more complex. In this case, the object texture is darker which imposes a higher aperture of the camera in order to perceive the pattern. This provokes that the rest of the scene is also visible. Nevertheless, a large set of spots are still decoded, including some of the projected on the background wall.

In all the examples, the decoding time was lower than 40 ms, which is the typical acquisition period of a CCIR format camera.



a) b) c)

Figure 2. Examples of decoded patterns when projected on different objects.

Next section reviews the typical definition of a positioning task by using visual data. In our case, the data used are the decoded image points provided by the coded structured light technique.

4. Visual servoing

As already said, a typical robotic task consists in positioning an eye-in-hand system with respect to an object by using visual features extracted from the camera. Visual servoing is based on the relationship between the camera motion and the consequent change on the visual features. This relationship is expressed by the well-known equation [5]

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v} \tag{2}$$

where $\mathbf{s}$ is a vector containing the visual features values, $\mathbf{L_s}$ is the so-called *interaction matrix*, and $\mathbf{v} = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$ the camera velocity screw.

The goal of visual servoing consists in moving the robot from an initial relative robot-object pose to a desired one where a desired set of visual features $\mathbf{s}^*$ is obtained. Most applications obtain the desired features $\mathbf{s}^*$ by using the *teaching-by-showing* approach. In this case, the robot is firstly moved to the desired position, then an image is acquired and $\mathbf{s}^*$ is computed. This is useful, for example, for target tracking and for robots having bad odometry, such as mobile robots. In these cases, the goal position can be achieved starting from the surroundings by using the visual servoing approach.

A robotic task can be described by a function which must be regulated to 0 [5]. Concretely, when the number of visual features is higher than the m degrees of freedom of the camera, the task function is noted as the following $m-$dimensional vector

$$\mathbf{e} = \widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{3}$$

where $\mathbf{s}$ are the visual features corresponding to the current state and $\widehat{\mathbf{L_s}}^{+}$ is the pseudoinverse of a model or an approximation of the interaction matrix. A typical control law for cancelling the task function and therefore moving the robot to the desired position is [5]

$$\mathbf{v} = -\lambda\mathbf{e} = -\lambda\widehat{\mathbf{L_s}}^{+}(\mathbf{s} - \mathbf{s}^*) \tag{4}$$

with λ a positive gain. It is well known that the local asymptotic stability of the control law is ensured if the model of interaction matrix holds

$$\widehat{\mathbf{L_s}}^{+}\mathbf{L_s} > 0 \tag{5}$$

As explained in the previous section, the coded pattern in Fig. 2 provides a large number of point correspondences in every image. Therefore, matching pattern points when viewing the object from different positions becomes straightforward. The normalised coordinates $\mathbf{x}$ of these points obtained after camera calibration can be used as visual features in the control loop. Given a set of k matched image points between the current and desired images, the visual features are defined by

$$\mathbf{s} = (x_1, y_1, x_2, y_2, ..., x_k, y_k) \tag{6}$$

The interaction matrix of a normalised point is [5, 6]

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{7}$$

where Z is the depth of the point. Then, $\mathbf{L_s}$ has the form

$$\mathbf{L_s} = \begin{bmatrix} -1/Z_1 & 0 & x_1/Z_1 & x_1y_1 & -(1+x_1^2) & y_1 \\ 0 & -1/Z_1 & y_1/Z_1 & 1+y_1^2 & -x_1y_1 & -x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -1/Z_k & 0 & x_k/Z_k & x_ky_k & -(1+x_k^2) & y_k \\ 0 & -1/Z_k & y_k/Z_k & 1+y_k^2 & -x_ky_k & -x_k \end{bmatrix} \tag{8}$$

Note that the real interaction matrix depends on the depth distribution of the points. Nevertheless, the depth distribution is usually considered as unknown and a rough approximation is used in the model of interaction matrix $\widehat{\mathbf{L_s}}$. A typical choice for $\widehat{\mathbf{L_s}}$, which is the one used in this paper,

is the interaction matrix evaluated at the desired state $\mathbf{L}_\mathbf{s}^*$. This is obtained by using the normalised coordinates from the desired image (x_i^*, y_i^*) and the depths in the desired position Z_i^*. In our case, the depths in the desired position have been modelled by setting $Z_i^* = Z^*$ being $Z^* > 0$ an approximation of the average distance between the object and the camera. Note, however, that other types of interaction matrix models could be used. For example, if the camera and the projector were accurately calibrated, the depth of the points could be reconstructed by using triangulation. Then, a better estimation of the interaction matrix in the desired state or even at each iteration would be available as will be shown in Section 6.

Another way to improve the system consists in considering alternative visual features computed from the 2D points, like image moments [19].

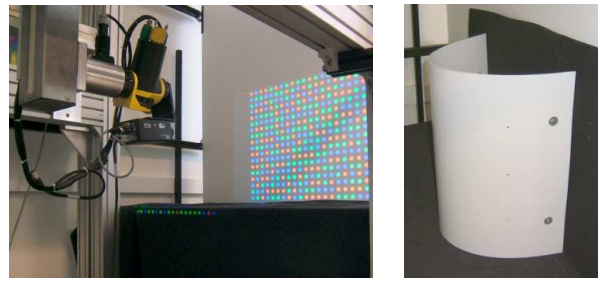
5. Experimental results

Experiments have been done in order to validate the visual servoing approach based on coded light. A robotic cell with a six-degrees-of-freedom arm has been used. A colour camera has been attached to the end-effector of the robot while a DLP projector has also been positioned about 1 m aside the robot. The projector focus has been set so that the pattern gets acceptably focused in a range of distances between 1.6 and 1.8 m with respect to the projector. This is the range of distances where the objects are placed during the experiments.

5.1. Planar object

The first experiment consists in positioning the robot with respect to a plane. Fig. 3a shows the robot manipulator and the plane with the encoded pattern projected on it. The desired position has been defined so that the camera is parallel to the plane at a distance of 90 cm. The reference image acquired in the desired position is shown in Fig. 4a. In this image a total number of 370 coloured spots out of 400 have been successfully decoded. The initial position of the robot in the experiment has been defined from the desired position, by moving the robot -5 cm along its X axis, 10 cm along Y, -20 cm along Z, and rotations of $-15°$ about X and $-10°$ about Y have been applied. The image perceived in this configuration is shown in Figure 4b. In this case, the number of decoded points is 361. Matching points between the initial and the desired images is straightforward thanks to the decoding process of the pattern.

The goal is then to move the camera back to the desired position by using visual servoing. At each iteration, the visual features set $\mathbf{s}$ in (6) is filled with the matched points between the current and the desired image. The normalised coordinates $\mathbf{x}$ of the points are obtained by using an approximation of the camera intrinsic parameters. The control law (4) is computed at each iteration with $\widehat{\mathbf{L}_\mathbf{s}} = \mathbf{L}_\mathbf{s}^*$. The result of the servoing is presented in Figure 4c-d. Concretely, the camera velocities generated by the control law are plotted in Figure 4c. Note that the norm of the task function decreases at each iteration as shown in Figure 4d. As can be seen, the behaviour of both the task function and the camera velocities is satisfactory and the robot reaches the desired position with no problem as for classical image-based visual servoing.



a) b)

Figure 3. a) First experiment: projection of the coded pattern onto a planar object. b) Elliptic cylinder used in the second experiment.

5.2. Non-planar object

In the second experiment a non-planar object has been used. Concretely, the elliptic cylinder shown in Figure 3b has been positioned in the workspace. In this case, the desired position has been chosen so that the camera points towards the object's zone of maximum curvature with a certain angle and the distance between both is about 60 cm. The desired image perceived in this configuration is shown in Figure 4e. The number of successfully decoded points is 160. Then, the robot end-effector has been displaced -20 cm along X, -20 cm along Y and -30 cm along Z. Afterwards, rotations of $-10°$ about X, $15°$ about Y and $5°$ about Z have been applied. These motions define the initial position of the robot end-effector for this experiment. The image perceived in this configuration is shown in 4f. In this case, the number of decoded points is 148.

The results of the visual servoing are plotted in Figure 4g-h. The desired image is reached again at the end of the servoing. Note that the model of interaction matrix used in the control law assumes that all the points ar coplanar at depth $Z^* = 60$ cm. Since the object has a strong curvature, the chosen model of the interaction matrix used is then very coarse. This explains that the camera velocities generated by the control law are more noisy and less monotonic than in the previous experiment. Furthermore, the convergence is slower. It has been proved that the depth distribution of a cloud of points used in classical image-based visual servoing plays an important role in the stability of the system [12]. Nevertheless, this experiment confirms that visual servoing is robust against modeling errors since the conver-

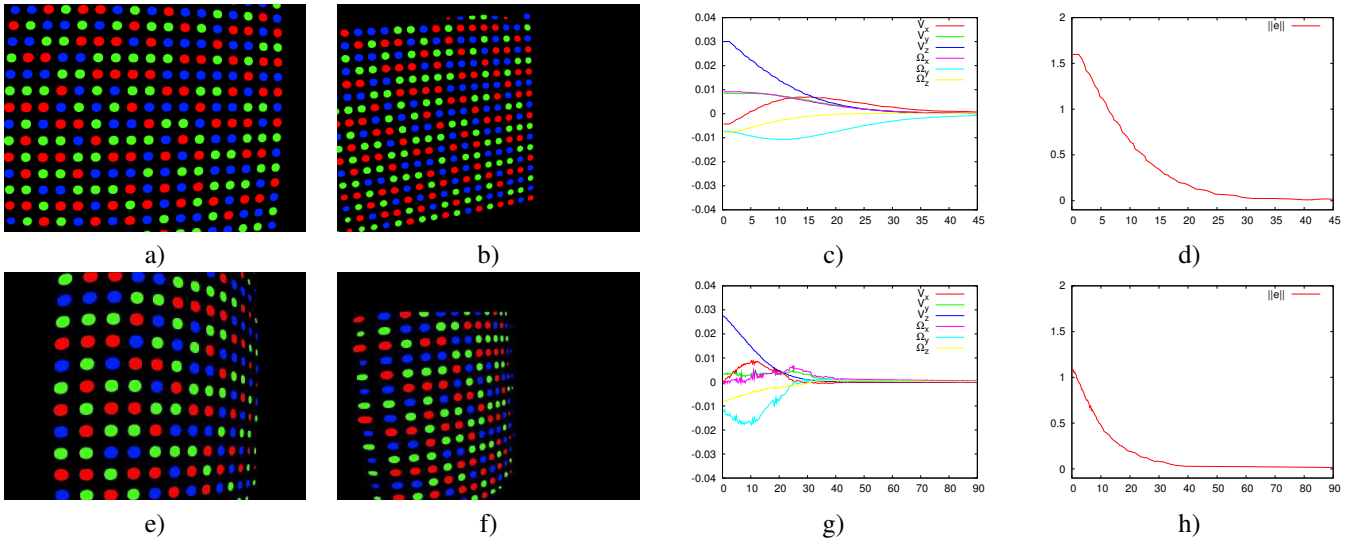


Figure 4. First experiment: planar object. a) Desired image. b) Initial Image. c) Camera velocities (ms/s and rad/s) vs. time (in s). d) Norm of the task function vs. time (in s). Second experiment: elliptic cylinder. e) Desired image. f) Initial Image. g) Camera velocities (ms/s and rad/s) vs. time (in s). h) Norm of the task function vs. time (in s).

gence is reached. In this experiment approximated camera intrinsic parameters are also used. Furthermore, during the robot motion, some of the pattern points were occluded by the robot arm. Therefore, the control law is robust against occlusions.

6. Advantages and limitations of the system

From the experiments, several advantages and limitations of the current system arise. First of all, a nice advantage of using the coded pattern is that the control law is robust to partial occlusions due to the large number of decoded points. This property is not achieved with systems projecting laser dots [15] as all the projected points are required. Another advantage is closely related to the large amount of points available. As shown in [6], the control law can be optimized by choosing those image points that minimize the condition number of the interaction matrix. It is well known that a good conditioning improves the stability of the system.

On the other hand, the current system has several limitations. First, the distance between the object and the robot is limited by the projector field-of-view. In principle, for a correct segmentation of the points, the pattern should be well-focused. However, since only the centers of gravity are needed, experiments show that even when the pattern is defocused the system converges. Another limitation appears when the projected dots are strongly distorted due to the object shape. In this case, the gravity centers cannot be precisely located. Nevertheless, these two modelling errors have shown poor effect on the convergence of the system. In fact, visual servoing is known to be pretty robust against modelling errors. The main limitation of the control law is the estimation of the depth distribution of the projected points. In the above experiments, the convergence has been reached by considering all the points as coplanar. However, with this rough approximation of the interaction matrix the convergence is not ensured in all the cases. In order to show the influence of the interaction matrix used in the control law some simulations are here presented. Three models of interaction matrix for the control law in (4) are tested:

- $\mathbf{L_s}(\mathbf{s}^*, \widehat{Z}^*)$: constant interaction matrix evaluated with the desired visual features and a rough approximation of the point depths $\widehat{Z}^*$. Concretely, all the points are assumed to be coplanar in the desired state. This is the choice made in the experiments of the previous section.

- $\mathbf{L_s}(\mathbf{s}^*, Z^*)$: constant interaction matrix evaluated with the desired visual features $\mathbf{s}^*$ and the real point depths Z^*.

- $\mathbf{L_s}(\mathbf{s}, Z)$: interaction matrix estimated at each iteration by using the current visual features $\mathbf{s}$ and the current depth Z of each point. Note that the dephts can be easily obtained with our system if a calibration process has been previously achieved as mentioned in Section 4.

The simulation consists in positioning a camera with respect to a cylindric object by using 5 points projected by a static emitter. A representation of the camera (blue), projector (red) and object (black) in the desired and initial configurations are shown in Figure 5a and Figure 5b, respectively.

The image points perceived by the camera in both configurations are shown in Figure 5c. The behavior of the system when using the control law based on $\mathbf{L_s}(\mathbf{s}^*, \widehat{Z}^*)$ is shown in Figure 5d and Figure 5g. In this case, in the desired state all the 5 points are considered to belong to a plane parallel to the image plane at a distance of $\widehat{Z}^* = 0.9$ m. As can be seen, in this example, contrary to the results obtained in the experiments, the system diverges.

In Figure 5e and Figure 5h we have the results obtained when using the control law based on $\mathbf{L_s}(\mathbf{s}^*, Z^*)$. In this case, the real depths of the points in the desired state are used. As can be seen, as this matrix is valid in a neighborhood around the desired state, the system is able to converge.

Finally, when using the control law based on $\mathbf{L_s}(\mathbf{s}, Z)$ the results are the ones in Figure 5f and Figure 5i. Note that as the real depths are used at each iteration the task is cancelled before. Therefore, a faster convergence is obtained. Furthermore, the kinematic screw of the camera is also better.

7. Conclusion

This paper has proposed an approach to visual servoing based on coded structured light for positioning eye-in-hand robots with respect to unknown objects. The projection of a coded pattern provides robust visual features independently of the object appearance. Furthermore, coded light enables to deal with non-textured objects or objects for which extracting visual features is too complex or too costly. Our approach is based on placing a DLP projector aside the robot manipulator. The use of a coded pattern allows classic visual servoing to be directly applied. This is advantageous since the large number of existing visual servoing techniques can be directly applied.

A pattern containing a M-array of coloured spots has been used for illustrating this approach. The choice of this pattern has been made taking into account its easy segmentation and fast decoding, which fits on the visual servoing requirements of sampling rate (40 ms). To our knowledge, this is the first work using coded structured light in a visual servoing framework. Therefore, we consider this approach a first step which shows the potentiality of coded light in visual servoing applications.

A classic image-based approach based on points provided by the coded pattern has been used. Experiments have shown that good results are obtained when positioning the robot with respect to planar objects. Furthermore, thanks to the large number of correspondences provided by the coded pattern, the system has shown to be robust even in presence of occlusions. On the other hand, the results when using non-planar objects show that the camera motion is noisier, slower and less monotonic. This is also a well known problem in classic 2D visual servoing when a rough estimation of the point depth distribution is included in the interaction matrix. In order to improve the results other existing image-based approaches could be tested like [3, 17, 19]. Furthermore, a better estimation of the depth distribution of the non-planar object would produce better results as shown through simulations.

Finally, we remark that structured light allows us to choose the visual features which will be used in the control law. Then, an important future work is to determine a suitable pattern design leading to a robust and optimised control law as done in [15] for the case of planar objects and onboard structured light. The use of an onboard emitter has many potential applications for example in mobile robots for target tracking and to reduce odometry errors.

References

[1] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. *Int. Journal of Robotics Research*, 21(8):679–700, August 2002.

[2] F. Chen, G. Brown, and M. Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 1(39):10–22, January 2000.

[3] P. I. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation*, 17(4):507–515, August 2001.

[4] C. J. Davies and M. S. Nixon. A hough transform for detecting the location and orientation of 3-dimensional surfaces via color encoded spots. *IEEE Trans. on systems, man and cybernetics*, 28(1):90–95, February 1998.

[5] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(6):313–326, June 1992.

[6] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell. Weighted selection of image features for resolved rate visual feedback control. *IEEE Trans. on Robotics and Automation*, 7(1):31–47, February 1991.

[7] P. Griffin, L. Narasimhan, and S. Yee. Generation of uniquely encoded light patterns for range data acquisition. *Pattern Recognition*, 25(6):609–616, 1992.

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

[9] R. A. Jarvis. A Perspective on Range Finding Techniques for Computer Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 5(2):122–139, 1983.

[10] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette. Visual servoing in robotics scheme using a camera/laser-stripe sensor. *IEEE Trans. on Robotics and Automation*, 12(5):743–750, 1996.

[11] F. J. MacWilliams and N. J. A. Sloane. Pseudorandom sequences and arrays. *Proceedings of the IEEE*, 64(12):1715–1729, 1976.

[12] E. Malis and P. Rives. Robustness of image-based visual servoing with respect to depth distribution errors. In *IEEE Int. Conf. on Robotics and Automation*, volume 1, pages 1056–1061, Taipei, Taiwan, September 2003.

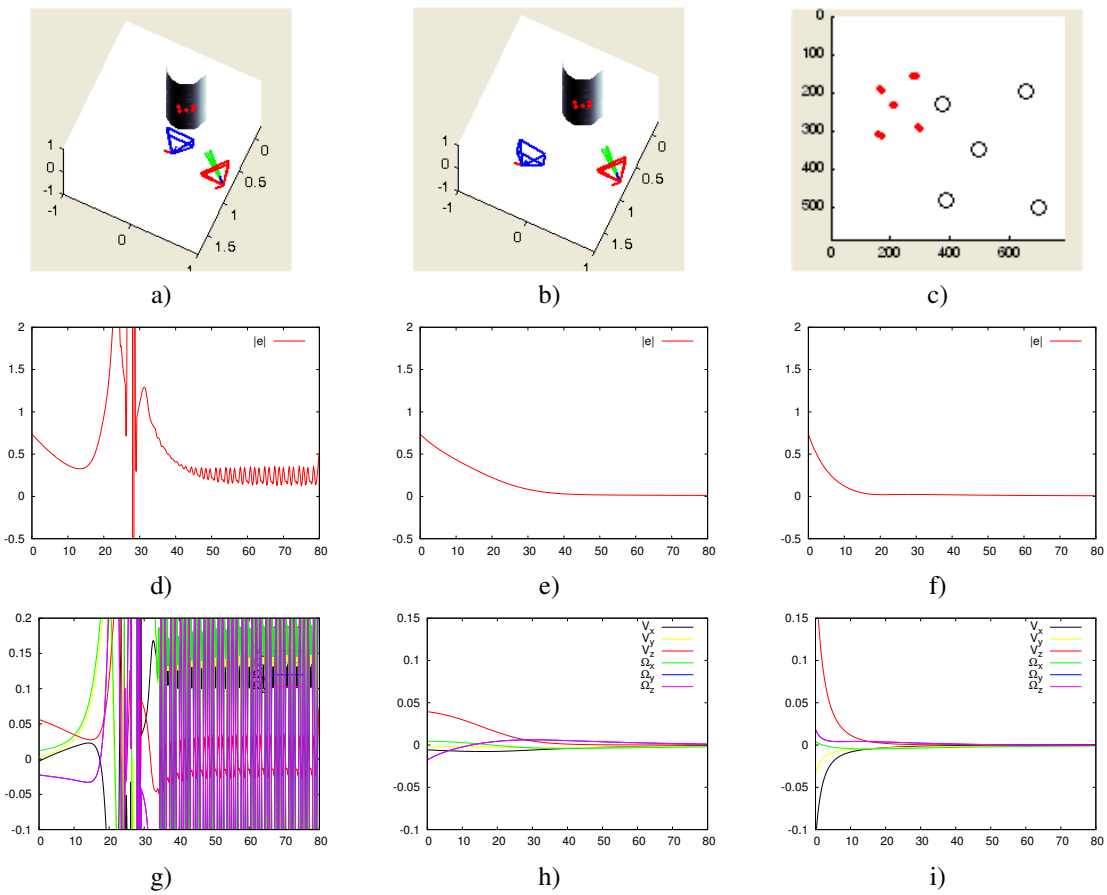


Figure 5. Simulations with a non-planar object. First row: a) Desired configuration b) Initial configuration c) Initial (red dots) and desired (circles) image points distributions. Second row: norm of the task function vs. time (in s) when using d) $\mathbf{L_s}(\mathbf{s}^*, \widehat{Z}^*)$ e) $\mathbf{L_s}(\mathbf{s}^*, Z^*)$ f) $\mathbf{L_s}(\mathbf{s}, Z)$. Third row: camera velocities (ms/s and rad/s) vs. time (in s). when using g) $\mathbf{L_s}(\mathbf{s}^*, \widehat{Z}^*)$ h) $\mathbf{L_s}(\mathbf{s}^*, Z^*)$ i) $\mathbf{L_s}(\mathbf{s}, Z)$.

[13] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissanov. Structured light using pseudorandom codes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):322–327, March 1998.

[14] J. Pagès, C. Collewet, F. Chaumette, and J. Salvi. Plane-to-plane positioning from image-based visual servoing and structured light. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 1004–1009, Sendai, Japan, September 2004.

[15] J. Pagès, C. Collewet, F. Chaumette, and J. Salvi. Robust decoupled visual servoing based on structured light. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 2676–2681, Edmonton, Canada, August 2005.

[16] J. Salvi, J. Pagès, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[17] F. Schramm, G. Morel, A. Micaelli, and A. Lottin. Extended-2d visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, pages 267–273, New Orleans, USA, April 26-May 1 2004.

[18] H. J. W. Spoelder, F. M. Vos, E. M. Petriu, and F. C. A. Groen. Some aspects of pseudo random binary array-based surface characterization. *IEEE Trans. on instrumentation and measurement*, 49(6):1331–1336, December 2000.

[19] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, 21(6), 2005.