

A New Method for Projector Calibration Based on Visual Servoing

J eremie Mosnier, Francois Berry and Omar Ait-Aider
 LASMEA - 24 avenue des landais - 63177 Aubi ere - France

jeremie.mosnier, francois.berry, omar.ait-aider@lasmea.univ-bpclermont.fr

Abstract

In this paper a new calibration method is proposed for projector using an uncalibrated camera. The difficulty to get a set of 2D-3D points in order to compute the projection matrix between the projector and the world has no easy solution. Methods are either inaccurate or inflexible. We propose an easy and automatic way to calibrate a projector. This method is a combination of pattern detection and visual servoing. The idea is to project a calibration pattern and superimpose it on a known printed pattern. We tested and demonstrated the feasibility and accuracy of our approach.

1 Introduction

Surface reconstruction is a very important topic due to the large field of applications. There are two classes of 3D scanners. First, we have passive scanners using principles like stereovision. Second, there are some active systems which use time-of-flight techniques and also structured light systems. Commonly a camera and a light source are used. This technique consists in projecting a pattern of light on a scene and acquiring images of that scene. The pattern is detected in the images and the 3D reconstruction can be done by triangulation. The coding strategy is really important in order to retrieve the information with good accuracy. A classification of these coding strategies has been done by Salvi et al.[1]. Three groups of patterns are defined: time-multiplexing, spatial neighborhood and direct coding. Time-multiplexing and direct codification are rather limited to static scene while Spatial neighborhood is focused on moving scenes. Readers could also refer to Ribo and Brandner[2] who gives a recent review of structured light techniques.

It is well known that one of the key points for the precision of such system depends on the quality of calibration. A lot of work has been done on the geometric model of that kind of system. The calibration process can be divided in two steps: the camera calibration then the projector calibration. The camera problem has been explored since a long time but the projector remains difficult to calibrate. Commonly two ways are proposed to solve the projector calibration task. The first one consists of estimating the coefficients of the equation of each light stripe plane. For instance Zhou and Zhang[3] proposed a method based on the observation of a planar target from at least two different

points of view. At least, 3 non-collinear world points are needed. The second way is to consider the projector as an inverse camera and then calibrate it in the same way. A set of 2D-3D points is used to estimate the perspective matrix of the projector. The most difficult step is to measure 3D points of the projected pattern in order to compute the perspective matrix. Some solutions to solve this problem are presented in the literature. Zhang and Huang[4] proposed a new concept of projector calibration. Indeed they enable the projector to "capture" images like a camera, this approach allows to calibrate the projector in the same way as the camera. Gao et al.[5] use a calibration board which is a paper sheet with circular control points printed on a laser printer. The paper is attached to a surface plane. A reference pattern which is a pattern with some horizontal red stripes and one green stripe is projected onto the surface plane by a LCD projector. Based on the cross ratio and the epipolar geometry, the 2D-3D point pairs for the projector and camera are obtained. In this way system is calibrated. Drouin et al.[6] proposed a large calibration pattern to take only one view using four cameras. They also define an energy formulation to simplify the process of finding correspondences. Recently, Drareni et al.[7] proposed a new calibration method for projector with a partially calibrated camera. They overcome the problem of determining the world-camera homography by searching the one which minimize the reprojection error. They demonstrates that a good accuracy is reached with fast time processing. Some solutions also exist to calibrate both devices (i.e. camera and projector) at the same time. For exemple Chen et al.[8] used a translating artefact of calibration in order to limit the number of views needed. They used a gray coded pattern to solve the matching problem.

However these methods are either inaccurate or need complex computation models. Our contribution is to propose an automatic technique which is easy to implement. Our calibration method does not consider epipolar geometry. We show that the calibration of the projector can be achieved in a simple way by using an uncalibrated camera and a known pattern. More precisely our method consists in projecting a calibration pattern with the projector and superimpose it on the printed pattern with the help of visual servoing. This allows to use the same 3D points (i.e. the world points of the printed pattern) for the camera and the projector and then considerably simplify the process

of videoprojector calibration.

The paper is structured as follows : in the second section the geometric model projector is described, in the third section we discuss our method to calibrate our system and finally we present some experimental results which validate our approach.

2 Projector Model

In this section, projector model (Fig.1) is described and several notations are specified.

In this approach, a classical pin-hole model is used [9]. Let us consider a point \mathbf{w} with the coordinates $(x^w y^w z^w)^T$ expressed by $\tilde{\mathbf{w}}$ in homogeneous coordinates. Let $\tilde{\mathbf{p}} = (x^p y^p 1)^T$ the projected point by the projector on the pattern. With a well-known formalism, the relation between \mathbf{p} and \mathbf{w} is given by the transformation matrix \mathbf{H}_p :

$$\tilde{\mathbf{p}} = \mathbf{H}_p \tilde{\mathbf{w}} = \mathbf{A}_p [\mathbf{R}_p \mid \mathbf{T}_p] \tilde{\mathbf{w}} \quad (2.1)$$

here \mathbf{H}_p is a 3×4 matrix, \mathbf{R}_p and \mathbf{T}_p respectively describe the 3×3 rotation matrix and the 3×1 translation vector. \mathbf{A}_p represents the intrinsic parameters of the projector such as:

$$\mathbf{A}_p = \begin{bmatrix} \alpha_{u_p} & \gamma_p & u_{0_p} \\ 0 & \alpha_{v_p} & v_{0_p} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

where $\alpha_{u_p} = -f.k_u$, $\alpha_{v_p} = -f.k_v$, $\gamma_p = f.k_u \cdot \cos \theta$, f is the focal length, k_u and k_v represent the number of pixels along the horizontal and vertical axes, θ is the angle between the u and v axes and (u_{0_p}, v_{0_p}) is the intersection between the optical axis and the image plane. The next section will present deeply our calibration process

3 Projector Calibration

The projector calibration matrix \mathbf{A}_p can be found in the same way as for the camera i.e. with a set of 2D-3D correspondences and a classical algorithm like

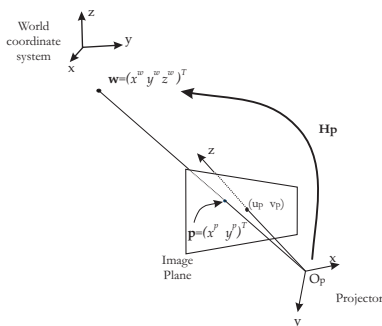


Figure 1: Projector model

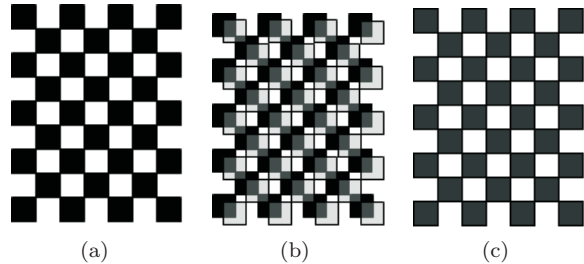


Figure 2: (a)printed pattern, (b)projected pattern partially fitted with printed pattern, (c)fitted projected pattern

[10] can be used. Compared to the camera calibration, the most difficult part is to get accurate 3D measurements. The existing solutions and their problems have been addressed in the introduction. Our contribution is to propose an easy calibration process by using a light pattern and a known laser printed pattern. The idea is to project a checkerboard printed pattern and fitting it with the known printed pattern (Fig.2(b) and 2(c)) by visual servoing. At the convergence, the projected pattern is fitted to the known printed pattern, and the matrix \mathbf{H}_p can be deduced. Indeed, it is well known under the term "perspective n -point problem" that given a set of at least 4 points with their coordinates in an object-centered frame and their corresponding projections onto an image plane, it is possible to find the transformation matrix \mathbf{H}_p between the object frame and the camera plane [11]. For that, we simulate a "perfect world" under OpenGL with a virtual camera and a virtual printed pattern. The virtual camera looks at the virtual printed pattern and displays it on the videoprojector. The 6 degrees of freedom of the virtual camera are controlled by the visual servoing in order to fit the projected pattern on the printed pattern. The following paragraph describes our system model.

3.1 System model

According to the previous explanation we can describe our model as in Fig.3. Let us consider first the camera case. A classical pin-hole model is used [9] but no calibration is performed. Only a weak calibration is needed, i.e. an estimation of focal length and optical center is given. This camera is used to detect features of the printed and projected pattern in the image. Knowing that features, we can easily compute the error in the image in order to process the visual servoing task (see next section). Then we deduce the velocity vector from this camera.

Concerning the projector, we showed in the previous section that we can modelize it like a pin-hole model. However, we cannot obviously apply the velocity vector directly to the projector as it is not moving. So we modelized the videoprojector as a virtual camera (Fig.4). Let us consider a point \mathbf{i} projected in a virtual world with the coordinates $(x^v y^v z^v)^T$ expressed by $\tilde{\mathbf{i}}$ in

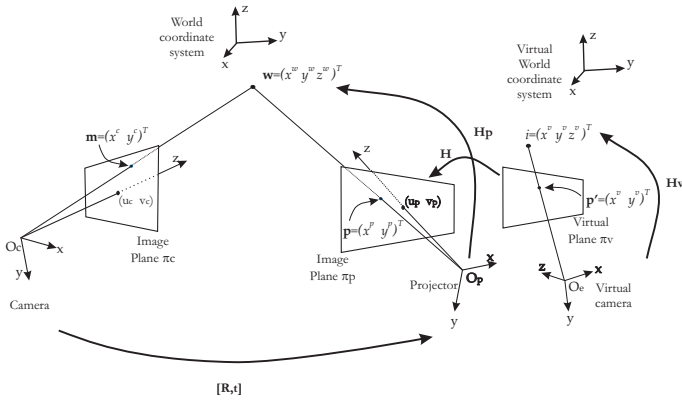


Figure 3: Global system overview

homogeneous coordinates. Let be $\tilde{\mathbf{p}}' = (x^v y^v 1)^T$ the projected point by the virtual camera on the virtual world. The relation between \mathbf{p}' and $\tilde{\mathbf{p}}'$ is given by the transformation matrix \mathbf{H}_v :

$$\tilde{\mathbf{p}}' = \mathbf{H}_v \tilde{\mathbf{p}} = \mathbf{A}_v [\mathbf{R}_v \mid \mathbf{T}_v] \tilde{\mathbf{p}} \quad (3.3)$$

The virtual plane is related with the projector as following:

$$\tilde{\mathbf{p}} = \mathbf{H}_p \tilde{\mathbf{p}}' \quad (3.4)$$

Where \mathbf{H} is the homography between the plans π_p and π_v . If the resolution of our virtual camera and our projector is the same, then this homography is set to identity. Thus the velocity vector is directly applied to the virtual camera and the projected pattern moves. The following paragraph explains the visual servoing bases and breaking down our approach.

3.2 Visual servoing

In the visual servoing approach [12], the control is directly specified in terms of regulation in the image. For a given task, a "target image" is built, corresponding to the desired position of the end effector with regard to the environment. It can be shown that all servoing schemes (Fig.5) are expressed like the regulation to zero of a function $\mathbf{e}(t)$ called task function and which is typically defined as

$$\mathbf{e}(t) = \mathbf{s}(m(t), a) - \mathbf{s}^* \quad (3.5)$$

where

- \mathbf{s}^* is considered as a reference target image to be reached in the image frame.
- $\mathbf{s}(m(t), a)$ is the value of visual information currently observed by the camera. $\mathbf{m}(t)$ is the image measurements and \mathbf{a} are the parameters representing potential additional knowledge of the system.

Considering a motionless target i.e., $\frac{d\mathbf{s}^*}{dt} = 0$ and an exponential decay of $\dot{\mathbf{e}}(t)$:

$$\dot{\mathbf{e}}(t) = -\lambda \mathbf{e}(t) \quad (3.6)$$

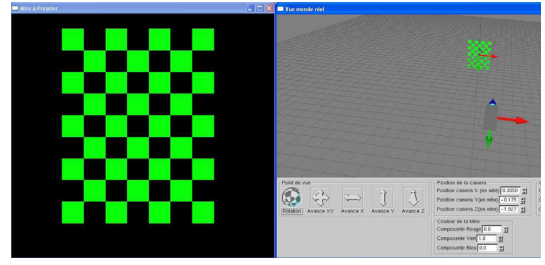


Figure 4: Software application under opengl

(λ is a positive scalar constant) a control law is given by :

$$\mathbf{T}_c = -\lambda \mathbf{L}^T e \quad (3.7)$$

where \mathbf{T}_c is the velocity screw of the camera and \mathbf{L}^T is image jacobian which characterizes the interaction between the sensor and its environment. The details of this matrix computation are well-known and can be easily found in [12].

In our case (Fig.6) a virtual camera is controlled in order to fit the projected pattern on the printed pattern. The vector \mathbf{s}^* corresponds to the visual features (corner positions) of the printed pattern. One reference image is taken in order to retrieve the 48 corners positions in the camera image representing the desired pose. The vector \mathbf{s} corresponds to corners extracted on the projected pattern (refer to section.3.3 for the pattern recognition method).

Thanks to these positions, the error and the interaction matrix are computed. Thus we deduce the velocity vector. However we also have to consider the rigid motion between the projector and the camera $[\mathbf{R}, \mathbf{t}]$ (Fig.1).

$$\mathbf{L}e = \mathbf{L} \begin{bmatrix} R & [t]_{\times} R \\ 0_3 & R \end{bmatrix} \quad (3.8)$$

Where $[t]_{\times}$ is the skew symmetric matrix associated with the translation vector t . The interaction matrix $\mathbf{L}e^{T+}$ is then used in the control law (Eq.3.7).

$$\mathbf{T}_c = -\lambda \mathbf{L}e^{T+} e \quad (3.9)$$

It has been shown that if we consider only a translation between the videoprojector and camera Eq.3.8 can be

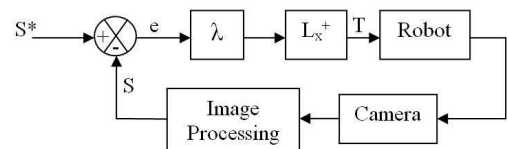


Figure 5: Classical visual servoing schema

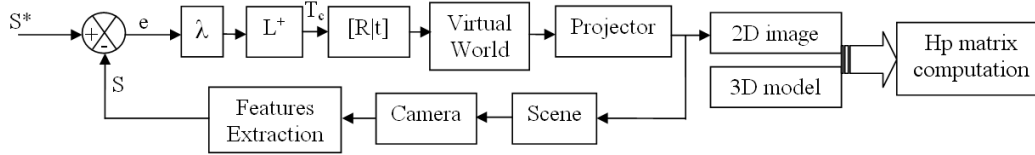


Figure 6: Calibration based on visual servoing

rewritten like:

$$\mathbf{L}\mathbf{e} = \mathbf{L} \begin{bmatrix} I & [t] \\ 0_3 & I \end{bmatrix} \quad (3.10)$$

Consequently the convergence $LL_e^+ > 0$ is immediately verified. However we need to take into account the movement imposed by the velocity vector (Fig.3). At each iteration the displacement is computed until the projected pattern fit the printed one. A 2D projector image is then synthesized.

These steps must be repeated to obtain different point of views (at least 10 times). Finally we obtain a set of 2D-3D points required to perform the calibration. The matrix \mathbf{H}_p is computed by SVD. Eq.2.1 can also be written as:

$$x^p = \frac{H_{p(11)}x^w + H_{p(12)}y^w + H_{p(13)}z^w + H_{p(14)}}{H_{p(31)}x^w + H_{p(32)}y^w + H_{p(33)}z^w + H_{p(34)}} \quad (3.11)$$

$$y^p = \frac{H_{p(21)}x^w + H_{p(22)}y^w + H_{p(23)}z^w + H_{p(24)}}{H_{p(31)}x^w + H_{p(32)}y^w + H_{p(33)}z^w + H_{p(34)}} \quad (3.12)$$

From equations 3.11 and 3.12, we can write the following linear system (with n image to scene points correspondences):

$$\mathbf{F}\mathbf{h}_p = 0 \quad (3.13)$$

With

$$\mathbf{h}_p = \begin{bmatrix} H_{p1}^T \\ H_{p2}^T \\ H_{p3}^T \end{bmatrix} \quad (3.14)$$

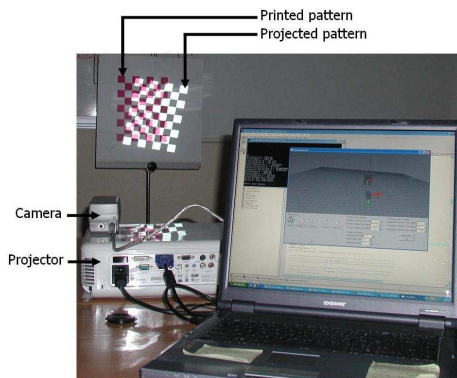


Figure 7: System configuration

Where $H_{p_i}^T$ is the i^{th} row of H_p . Thus the vector \mathbf{h}_p can be found by computing the SVD of matrix \mathbf{F} . Noting $\mathbf{F} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, the last column of \mathbf{V} corresponding to the smallest singular value of \mathbf{F} is the solution of Eq.3.14. Once \mathbf{H}_p is known the matrix \mathbf{A}_p is found by QR Decomposition.

3.3 Pattern recognition using colorspace

One of the key points of our approach is to distinguish the projected and printed pattern. In order to perform this task different colors are used for each pattern. Fig.2 illustrates the interest of color to identify the two different patterns. Two methods have been tested for the detection of interest points. First using a reference image (a picture with the printed pattern only) and an image with the projected pattern. Second other using just one image. Obviously, the first method leads to better results. Without the reference, the differentiation of the two patterns is more difficult and the system is more sensitive to luminosity problem. Concerning the feature extraction, the thresholding used is tuned to recognize different colors.

4 Experiments

4.1 Hardware

Our system (Fig.7) works with a color camera SONY DFW-VL500 which uses the IEEE 1394 protocol. Its maximum resolution is 640×480 pixels. The video projector is a Optoma VPL-CX75. Its maximum resolution is 1600×1200 pixels.

4.2 Software

We have tested our system on two different softwares. One uses Matlab©, the calibration is done by the toolbox [13]. The second one is written in C++. The calibration is developed with opencv which proposes some dedicated tools. The visual servoing task and the creation of the projected pattern has been developed in C++ with the opengl library (Fig.4). The camera image acquisition is also coded in C++ and all the system is running on windows xp©.

4.3 Results

An example typical visual servoing process is given in Fig.8. We can observe on this image the starting

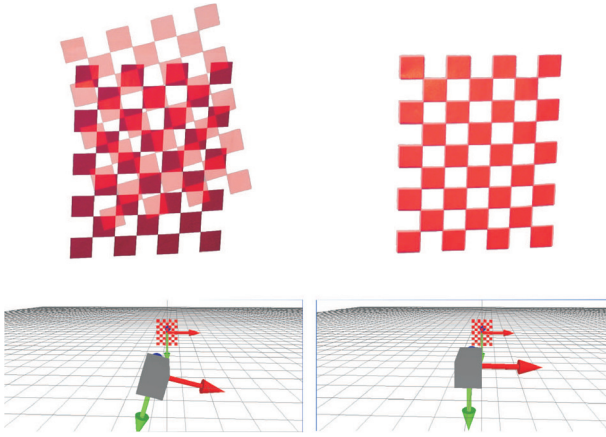


Figure 8: Fitting process example. Left :the starting position of the projected pattern with its virtual representation. Right : Projected pattern totally fitted with the printed one

position of the projected pattern and then the result when it is totally fitted on the printed one. We must repeat this process at least 10 times with different orientations of the printed pattern. Thus, the calibration is performed. To verify the integrity of the calibration procedure we measured a planar checkerboard and retrieved the intrinsic and extrinsic parameters of the projector. The results of the intrinsic parameters are presented in Table.1. We also computed the reprojection error using the obtained intrinsic parameters of the projector. (Fig.9) shows the distribution of this error. The mean value was 0.20715 and the standard deviation was 0.1836. Our results are good as comparison to other methods.

Table 1: Projector intrinsic parameters

f_x	f_y	u_0	v_0
2074.19366	2075.97916	812.20177	622.27101

5 Conclusion and Future Work

We have proposed a new of calibration method for structured light system based on visual servoing. This technique simplifies the process of calibration as the same 3D points are used for both camera and projector. The experimental results prove the accuracy of the method. In the future, more tests (in terms of 3D reconstruction) should be performed in order to compare more precisely our method with other ones.

Acknowledgement

We would like to thank the "Pôle Innovande" of Clermont Ferrand, France for the financial support for this work(Project SRDVIand).

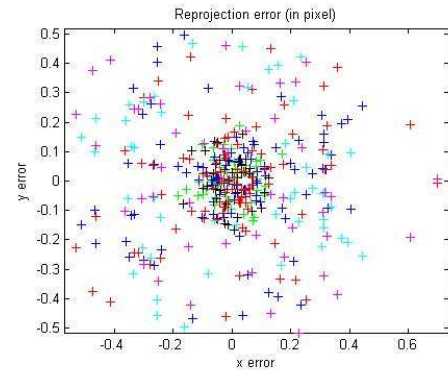


Figure 9: Reprojection error

References

- [1] J. Salvi, J. Pages, and J. Battle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(1):827–849, 2004.
- [2] M. Ribo and M. Brandner. State of the art on vision-based structured light systems for 3d measurements. *Robotics and Sensor Environments*, pages 2–6, 2005.
- [3] F. Zhou and G. Zhang. Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations. *Image and Vision Computing*, 1:59–67, 2005.
- [4] S. Zhang and P.S. Huang. Novel method for structured light system calibration. *Optical Engineering SPIE*, 45, 2006.
- [5] W. Gao, L. Wang, and Z. Hu. Flexible calibration of a portable structured light system through surface plane. *Acta Automatica Sinica*, 2007.
- [6] M.-A Drouin, G. Godin, and S. Roy. An energy formulation for establishing the correspondence used in projector calibration. *3D Data Processing, Visualization and Transmission*, pages 18–20, 2008.
- [7] Jamil Drarni, Peter Sturm, and Sébastien Roy. Projector calibration using a markerless plane. In *Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal*, volume 2, pages 377–382, feb 2009.
- [8] X. Chen, J. Xi, Y. Jin, and J. Sun. Accurate calibration for a camera-projector measurement system based on structured light projection. *Opt Laser Eng*, 2008.
- [9] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, 2000.
- [10] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. *ICCV*, pages 666–673, 1999.
- [11] Radu P. Horaud, Bernard Conio, Olivier Le Boulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics and Image Processing*, 47:33–44, 1989.
- [12] S. Chaumette and S. Hutchinson. Visual servo control part i: Basic approaches. *Robotics & Automation Magazine*, 13:82–90, 2006.
- [13] Jean-Yves Bouguet. Calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/.