

Model-free augmented reality by virtual visual servoing

Muriel Pressigout, E. Marchand

▶ **To cite this version:**

Muriel Pressigout, E. Marchand. Model-free augmented reality by virtual visual servoing. IAPR Int. Conf. on Pattern Recognition, ICPR'04, 2004, Cambridge, UK, United Kingdom. pp.887-891. inria-00352035

HAL Id: inria-00352035
https://inria.hal.science/inria-00352035

Submitted on 12 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.

IAPR Int. Conf. on Pattern Recognition, ICPR'04, Cambridge, UK, August 2004.

Model-free augmented reality by virtual visual servoing

Muriel Pressigout, Éric Marchand
IRISA INRIA Rennes
Campus de Beaulieu, 35042 Rennes Cedex, France
E-mail: Firstname.Lastname@irisa.fr

Abstract

This paper presents a method based on the virtual visual servoing approach [10] to achieve markerless augmented reality applications. This work aims to realize this task using as little prior 3D information as possible. Virtual visual servoing techniques that lead to a non-linear minimization approach allow one to estimate the 2D transformation between two images of a video sequence which permits to achieve augmented reality on this sequence. Thanks to the work that has already been carried out in this domain, the presented method is efficient and robust wrt. noise and occlusions. It allows very realistic augmented videos with minimum knowledge about the real environment.

1. Introduction

Augmented reality (AR) [1] aims to insert virtual objects in a real environment captured by a moving camera, in a manner such that these objects seem to be part of the viewed 3D scene. This work is related to the AR problem in the case of an unique camera. The most important issue is to overcome the registration problem, *i.e.* how to align the real and the virtual world properly to give the impression that they are just one world. In a vision-based system, this is usually a pose computation issue.

Most of the approaches consider the pose computation as a registration problem that consists of determining the relationship between 3D coordinates of features (points, lines,...) and their 2D projections onto the image plane [2, 3, 9]. These approaches require a 3D scene model obtained by fiducial markers or by exploiting its structure. Since such 3D knowledge is not easily available, it is necessary to overcome the pose computation considering less constraignant knowledge on the viewed scene. This can be done by using planar structures of the scene [8, 13, 12]. Whatever the method chosen, it must deal with the problem of robustness to account for the noise and occlusion phenomenons it may include since the content of the video is unknown.

This work copes with the 3D knowledge issue by using, at most, the 2D information extracted from the images and the geometrical constraints inherent to a moving vision system [5]. It has been chosen to estimate the camera displacement between the capture of two images instead of the camera pose. This can be accurately achieved by minimizing a distance in the image defined using the strong constraints linking two images of the same scene. The novelty of this article is that the camera displacement estimation by a non-linear minimization is considered like a problem of 2D virtual visual servoing (VVS) [10]. It is therefore closer to the underlying geometrical constraints than similar classical approaches as described in, *e.g.* , [5].

This article first describes how the displacement estimation can be handled like a problem of 2D VVS and then how it can be made robust. The following sections set out the different displacement cases we dealt with and how to use the displacement estimate for AR with minimum prior 3D knowledge. Finally, several experimental results on real videos are presented.

2. Computing Displacement

As already stated, the fundamental principle of the proposed approach is to define a non-linear minimization approach as the dual problem of 2D visual servoing [7]. This formulation has already been applied to the pose computation problem [2, 10]. In visual servoing, the goal is to move a camera in order to observe an object at a given position in the image. This is achieved by moving the camera in order to minimize the error between a desired state of the image features $\mathbf{s}^*$ and the current state $\mathbf{s}$. Displacement computation problem is a very similar issue.

To illustrate the principle, consider the case of a scene with various 2D features $\mathbf{s}$ (for example, points, distances,...). For camera motion estimation the classical idea is to minimize the distance between the position of the observed features in image 2 ($\mathbf{s}_2$) and their position ${}^2tr_1(\mathbf{s}_1)$ transfered in the image 2 by a given transformation (represented by the fundamental or essential matrix, an homography, etc...) whose parameters rely on the camera displace-

ment ${}^2\mathbf{T}_1$ to be estimated:

$$\widehat{{}^{c_2}\mathbf{M}_{c_1}} = \arg\min_{{}^{c_2}\mathbf{M}_{c_1}} \Delta \text{ with } \Delta = \sum_{i=1}^{N} d(\mathbf{s}_{2_i}, {}^2tr_1(\mathbf{s}_{1_i}))$$

In this formulation of the problem, a virtual camera is moved (initial displacement is null) using a visual servoing control law in order to minimize this error Δ. At convergence, the virtual camera reaches the position ${}^{\mathbf{2}}\mathbf{M}_{\mathbf{1}}{}^*$ which minimizes this error (${}^{\mathbf{2}}\mathbf{M}_{\mathbf{1}}{}^*$ will be the real camera displacement). It is supposed in this paper, that intrinsic parameters are available.

In the more realistic case where image measurement errors occur in both images, it is better to minimize the errors in both images and not only in one. We then have to consider the forward (${}^2tr_1$) and backward (${}^1tr_2$) transformation. The distance to be minimized is then :

$$\sum_{i=1}^{N} d(\mathbf{s}_{2_i}, {}^2tr_1(\mathbf{s}_{1_i})) + d(\mathbf{s}_{1_i}, {}^1tr_2(\mathbf{s}_{2_i})) \quad (1)$$

where N is the number of considered features and $d(\mathbf{s}_{2_i}, {}^2tr_1(\mathbf{s}_{1_i})) = {}^2d_{1i}$ is the signed distance between the 2D features $\mathbf{s}_{2_i}$ and ${}^2tr_1(\mathbf{s}_{1_i})$. Minimizing this distance is equivalent to minimize the error vector :

$$\mathbf{e} = \left(\ldots, {}^2d_{1i}, {}^1d_{2i}, \ldots\right)^T$$

by the following control law :

$${}^{\mathbf{2}}\mathbf{v} = -\lambda \widehat{\mathbf{L}}^{+}\mathbf{e} \quad (2)$$

where ${}^{\mathbf{2}}\mathbf{v}$ is the velocity of the virtual camera (expressed in camera 2 frame) and where $\mathbf{L}$ is the interaction matrix related to the error vector such as :

$$\widehat{\mathbf{L}} = \left(\cdots, \quad \widehat{\mathbf{L}}({}^2d_{1i}), \quad -\widehat{\mathbf{L}}({}^1d_{2i}){}^1\widehat{\mathbf{V}}_2, \quad \cdots \right)^T \quad (3)$$

$\mathbf{L}({}^2d_{1i})$ is the Jacobian matrix that links the variation of the distance ${}^2d_{1i}$ to the virtual camera velocity such as : ${}^2\dot{d}_{1i} = \mathbf{L}({}^2d_{1i}){}^{\mathbf{2}}\mathbf{v}$. We will see how to define this matrix in section 2.1. ${}^1\mathbf{V}_2$ is the velocity transformation matrix from camera 1 frame to camera 2 frame, given by the following 6×6 matrix:

$${}^1\mathbf{V}_2 = \begin{pmatrix} {}^1\mathbf{R}_2 & [{}^1\mathbf{t}_2]_\times {}^1\mathbf{R}_2 \\ \mathbf{0}_{3\times 3} & {}^1\mathbf{R}_2 \end{pmatrix}$$

where $[\mathbf{t}]_\times$ is the skew matrix related to the vector $\mathbf{t}$.

As shown in [2], if data are corrupted with noise, the widely accepted statistical techniques of robust M-estimation [6] can be introduced within the minimization process. This is introduced directly in the virtual visual servoing control law by weighting the confidence on each feature.

$${}^{\mathbf{2}}\mathbf{v} = -\lambda(\widehat{\mathbf{D}}\widehat{\mathbf{L}})^{+}\widehat{\mathbf{D}}\mathbf{e} \quad (4)$$

where $\mathbf{D}$ is a diagonal weighting matrix given by $\mathbf{D} = diag(..., w, ...)$ The weights w_i reflect the confidence of each feature. Their computation needs an influence function. Tukey's hard re-descending function is considered since it completely rejects outliers and gives them a zero weight (see [2, 6] for further information on weights computation and influence functions). This is of interest in this sort of application so that a detected outlier has no effect on the virtual camera motion.

2.1. General camera motion

This subsection describes the 2D transformation to be estimated for the most general case: a non-planar scene viewed by a camera which rotates and translates. In the reminder of the paper features we use the following notation: $\mathbf{p}_1$ for the points extracted from camera 1 image and $\mathbf{p}_2$ for the corresponding points in camera 2 image. In that case the constraints derived from the epipolar geometry give [5] :

$$\mathbf{p}_1^T {}^1\mathbf{E}_2\mathbf{p}_2 = 0 \text{ and symmetrically } \quad \mathbf{p}_2^T {}^2\mathbf{E}_1\mathbf{p}_1 = 0 \quad (5)$$

The 3×3 matrix ${}^2\mathbf{E}_1 = [{}^1\mathbf{t}_2]_\times {}^1\mathbf{R}_2$ is called the essential matrix. ${}^2\mathbf{E}_1$ is only related to the camera displacement and is the same for all the considered 3D points. In this case computing the camera motion is equivalent to compute this essential matrix. Considering the virtual visual servoing approach the idea is to minimize the distance between the position of the observed points in image 2 ($\mathbf{p}_2$) and the position of the corresponding features ${}^2tr_1\mathbf{p}_1$ transfered in the image 1 by the essential matrix ${}^2\mathbf{E}_1$, *i.e.* to minimize the signed difference between $\mathbf{p}_2$ and their associated epipolar lines $\mathbf{l}_2$ in the image i. Hence, the terms of the global error $\mathbf{e}$ (2) to be minimized in both image 1 and 2 are obtained by :

$${}^2d_{1i} = \mathbf{p}_{2_i}^T \mathbf{l}_{1i} \text{ and } {}^1d_{2i} = \mathbf{p}_{1_i}^T \mathbf{l}_{2i} \quad (6)$$

(6) means that a point $\mathbf{p}_1$ must rely on the epipolar line $\mathbf{l}_1$ related to its corresponding point $\mathbf{p}_2$ such as $\mathbf{l}_1$ is defined by ${}^1\mathbf{E}_2\mathbf{p}_2$. The epipolar line $\mathbf{l}_2$ line related to $\mathbf{p}_1$ is the projection of the line $\mathbf{C}_1\mathbf{P}$ (where $\mathbf{C}_1$ is the camera optical center and $\mathbf{X}$ is the 3D point that project in $\mathbf{p}_1$ and $\mathbf{p}_2$).

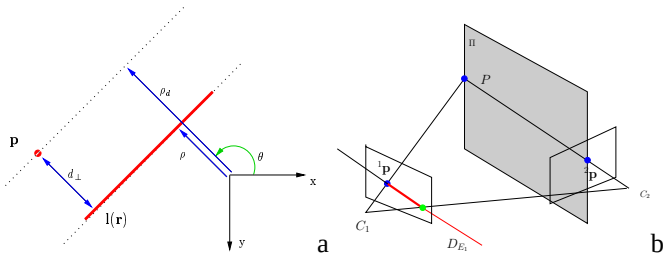


Figure 1. (a) Distance of a point to a line. (b) Plane Π used in the computation of the interaction matrix

The distance between point $\mathbf{p}$ and line $\mathbf{l}(r)$ can be characterized by the distance $d_\perp$ perpendicular to the line. Thus the distance feature from a line is given by:

$$d_l = d_\perp(\mathbf{x}, \mathbf{l}(r)) = \rho(\mathbf{l}(r)) - \rho_\mathbf{p} \tag{7}$$

where $\rho_\mathbf{p} = x\cos\theta + ysin\theta$, with x and y being the coordinates of the tracked point. Thus,

$$\dot{d}_l = \dot{\rho} - \dot{\rho_\mathbf{p}} = \dot{\rho} + \alpha\dot{\theta}, \tag{8}$$

where $\alpha = x\sin\theta - y\cos\theta$. Deduction from (8) gives $\mathbf{L}_{d_l} = \mathbf{L}_\rho + \alpha\mathbf{L}_\theta$. The interaction matrix related to d_l can be thus derived from the interaction matrix related to a straight line given by (see [4] for its complete derivation):

$$\begin{array}{l}\mathbf{L}_\theta = \begin{pmatrix} \lambda_\theta\cos\theta & \lambda_\theta\sin\theta & -\lambda_\theta\rho & \rho\cos\theta & -\rho\sin\theta & -1 \end{pmatrix} \\ \mathbf{L}_\rho = \begin{pmatrix} \lambda_\rho\cos\theta & \lambda_\rho\sin\theta & -\lambda_\rho\rho & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \end{pmatrix}\end{array} \tag{9}$$

where $\lambda_\theta = (A_2\sin\theta - B_2\cos\theta)/D_2$, $\lambda_\rho = (A_2\rho\cos\theta + B_2\rho\sin\theta + C_2)/D_2$, and $A_2X + B_2Y + C_2Z + D_2 = 0$ is the equation of a 3D plane Π which the line belongs to (see Figure 1b).

The translation ${}^1\mathbf{t}_2$ is estimated up to scale. Indeed, if the displacement between the image 1 and the image 2 such as the translation is ${}^1\mathbf{t}_2$ and the rotation ${}^1\mathbf{R}_2$ obeys to (5), so does a similar one such as ${}^1\mathbf{t}_2{}' = k.{}^1\mathbf{t}_2$ and ${}^1\mathbf{R}_2{}' = {}^1\mathbf{R}_2$. In order to find the exact translation, 3D information is needed. It can be a distance between two points of the scene: there is only one scalar k that keeps constant this 3D distance such as the real translation and rotation are respectively $k.{}^1\mathbf{t}_2$ and ${}^1\mathbf{R}_2$.

2.2. Homography estimation

Some particular cases of camera displacement (planar scene, pure rotation camera motion) leads the 2D transformation between two images of the video to be a homography. In that case, this gives:

$$\mathbf{p}_2 = {}^2\mathbf{H}_1\mathbf{p}_1 = ({}^i\mathbf{R}_j + \frac{{}^i\mathbf{t}_j}{{}^jd}{}^j\mathbf{n}^T)\mathbf{p}_1 \tag{10}$$

where ${}^2\mathbf{H}_1$ is an homography that defined the transformation between the image acquired by the camera 1 and the camera 2. In this case computing the camera motion is equivalent to compute this homography. When considering the virtual visual servoing approach the idea is to minimize the distance between the position of the observed points in image 2 ($\mathbf{p}_2$) and the position of the corresponding points $\mathbf{p}_1$ transfered in the image 2 by the homography ${}^2\mathbf{H}_1$. The goal is then to minimize the error (2) in both image 1 and 2 whose terms are given by:

$${}^2d_{1i} = {}^2\widehat{\mathbf{H}}_1\mathbf{p}_{1i} - \mathbf{p}_{2i} \text{ and } {}^1d_{2i} = {}^2\widehat{\mathbf{H}}_1^{-1}\ \mathbf{p}_{2i} - \mathbf{p}_{1i}$$

The terms $\mathbf{L}({}^jd_{ki})$ of related interaction matrix $\mathbf{L}$ are thus the classical interaction matrix that links the variation of the point $\mathbf{x}_i$ position to the camera motion (see *e.g.* [7]).

3. Application to augmented reality

For augmented reality applications, the pose between the camera and the world coordinate system is required. If an initial pose ${}^1\widehat{\mathbf{M}}_W$ is known, computing the current pose from the estimated displacement is straightforward:

$${}^n\widehat{\mathbf{M}}_W = {}^n\widehat{\mathbf{M}}_1{}^{n-1}\widehat{\mathbf{M}}_W \tag{11}$$

since the displacement between the first and the current image is computed, using the precedent image displacement estimation as initial estimation. However computing ${}^1\widehat{\mathbf{M}}_W$ requires the introduction of 3D information. Therefore it has been decided to estimate this first pose from the image of a rectangle in the first image following the approach presented in [13]. The only 3D information required is a rectangle in the first image and the lenght of one of its sides.

4. Experimental results

For the outdoor experiments, tracking is achieved along the image sequence using the Shi-Tomasi-Kanade points tracker [11].

4.1. General case: estimating the essential matrix

In this first experiment, the camera undergoes a translation and a rotation. There are some markers in the viewed scene that allow a fast tracking and provide a reliable set of points in each image of the video sequence without any matching problem. The 3D information used is a rectangle extracted from the markers in the initial image to compute the initial pose and the lenght of one of its sides during the sequence to estimate the right translation. In Figure 2, three augmented images of this sequence are shown. One can see that the added horse remains at the same location along the sequence.



Figure 2. AR from general camera motion

4.2. Planar scene: estimating the homography

In this experiment (see Figure 3), an outdoor scene is considered. The wall is the planar scene from which points are extracted to estimate the homography between two images. The rectangle used to estimate the initial pose is the one composed by the different posters. It is not very accurate but it provides a good enough result. The pose computation resulting from this initial pose estimation and the displacement estimations provide realistic augmented video sequence as can be seen in Figure 3. The objects remain stable in the scene.



Figure 3. AR with robust homography estimation with planar structure

Two comparaisons have been made on the remaining error between the image points and the projection of the corresponding points in the other image for the estimated displacement (see the Figure 4). The presented method is first compared using the robust kernel and without. It can be noticed that after a while, the use of M-estimator gives really better displacement estimations. It is then compared with the linear one, *i.e.* the DLT algorithm using the data normalisation recommended by [5]. It is undeniable that the presented method, even without its robust kernel, is far more efficient. However other non-linear minimization approaches give similar results.

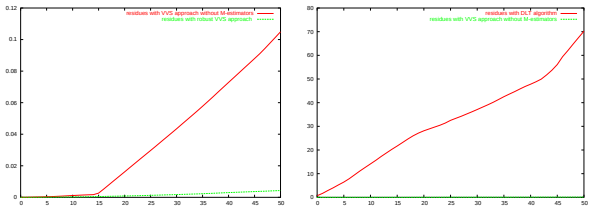


Figure 4. Planar structure. Left: VVS without M-estimators (red) *vs.* **robust VVS (green). Right: DLT algo (red)** *vs.* **VVS without M-estimators (green).**

4.3. Pure rotation camera motion

Pure rotation is interesting since in this case the homography ${}^{i}\mathbf{H}_{j}$ is only related to the rotation. Thus the points are not required to belong to a plane. This particularity may be considered in a lot of image sequences where the camera translations are very small. The equations presented for homography estimation have been simplified by removing the terms related to the motion translation. In this experiment (see Figure 5), an outdoor scene is considered with very noisy images. The Figure 5 shows that even after 800 images, the error in pose computation (thus in displacement computation) is very small. What must be pointed out is that the complete change of background during the sequence does not disturb the results.

5. Conclusion

This paper shows that exploiting the virtual visual servoing approach to achieve displacement estimation based on 2D information is efficient and furthermore is intuitive since it is nearer to the underlying geometrical constraints than the other non-linear minimization approaches. Robust estimation is obtained by the introduction of the M-estimators in the control law which updates the displacement estimation. Its application to AR provides very realistic videos with very few constraints.

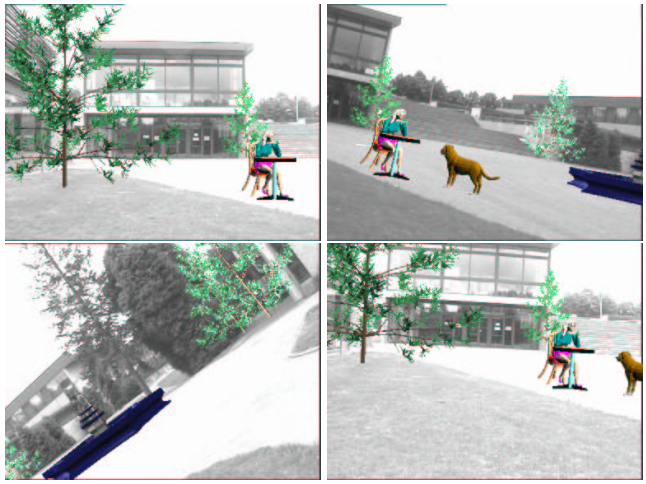


Figure 5. AR with pure rotation camera motion

Aknowledgment This work was realized in the context of the french RIAM Sora project in Lagadic team at IRISA/INRIA Rennes.

Videos are available on the Lagadic website: **http://www.irisa.fr/lagadic**

References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre. Recent advances in augmented reality. *IEEE CG&A*, 21(6):34–47, 2001.

[2] A. Comport, E. Marchand, F. Chaumette. A real-time tracker for markerless augmented reality. In *IEEE/ACM ISMAR*, pp. 36–45, 2003.

[3] D. Dementhon, L. Davis. Model-based object pose in 25 lines of codes. *IJCV*, 15:123–141, 1995.

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

[5] R. Hartley,A. Zissermann. *Multiple View Geometry in computer vision*. Cambridge Univ. Press, 2001.

[6] P.-J. Huber. *Robust Statistics*. Wiler, New York, 1981.

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

[8] K. Kutulakos, J. Vallino. Calibration-free augmented reality. *IEEE TVCG*, 4(1):1–20, 1998.

[9] D. Lowe. Fitting parameterized three-dimensional models to images. *IEEE PAMI*, 13(5):441–450, 1991.

[10] E. Marchand, F. Chaumette. Virtual visual servoing: a framework for real-time augmented reality. In *EUROGRAPHICS*, volume 21(3), pp. 289–298, 2002.

[11] J. Shi, C. Tomasi. Good features to track. In *IEEE CVPR*, pp. 593–600, 1994.

[12] G. Simon, M.-O. Berger. Pose estimation for planar structures. *IEEE CG&A*, 22(6):46–53, 2002.

[13] G. Simon, A. Fitzgibbon, A. Zisserman. Markerless tracking using planar structures in the scene. In *IEEE/ACM ISAR*, pp. 120–128, 2002.