Applications and Specificities of Synthetic/Synthetic Projective Registration

Emmanuel Garcia and Jean-Luc Dugelay

Institut Eurécom, Multimedia Department
2229 route des Crêtes, 06904 Sophia Antipolis, France
{garciae, dugelay}@eurecom.fr

Abstract

A recent application related to 3D watermarking has led to a specific registration problem: the registration of a 3D computer object with a computer-generated 2D view of it. So far, projective registration algorithms have focused on images of real objects because there was no interest in registering synthetic images with computer models. While those algorithms could also be directly applied to the case of synthetic images, they do not take advantage of some specificities of the synthetic/synthetic registration problem. This problem is addressed here and a dedicated registration algorithm is presented.

1. Introduction

The problem of projective registration consists in finding the parameters of the perspective projection between a 3D object and a 2D view of it. It shows up, directly or indirectly, in most 3D computer vision problems: camera calibration, model-based tracking, 3D reconstruction, or texturing of a 3D model [5], to cite a few.

These computer applications usually act as a link between the physical world and a computer representation of it. Their inherent limitation lies in the difficulty to model the physical world with a tractable and arbitrarily accurate model.

On the other hand, there exists another kind of problems involving projective registration that have quite different properties and requirements. These problems involve registering a computer 3D object with a *computer-generated* 2D view of this object. The fundamental difference of this kind of problems is that by construction an exact 3D model and a perfectly accurate projection model that lead to the 2D image to be registered are guaranteed to exist.

So far, no computer vision algorithm takes advantage of the specific properties of synthetic/synthetic projective registration because there are virtually no applications that involve it. However such an application has been identified recently: texture-based watermarking of 3D video objects [2]. This application requires a high accuracy of the projective registration. To achieve this a specific algorithm must be developed.

In section 2 we recall some principles of projective registration in computer vision. In section 3 we briefly present the texture-based watermarking application and the specificities of the projective registration involved. In section 4 we describe a projective registration algorithm for synthetic images of textured 3D object and we provide the first results in section 5.

2. Projective registration and computer vision

In the context of computer vision the challenge consists in registering a computer model of a real object with an image of it captured by a camera. The relation between the model of the object and its image may be approximated by a perspective projection. This approximation may not be very accurate in the case of a camera that has significant lens distortion. The projection model may be made more accurate by modeling and estimating the lens distortion factor as a part of the projective registration process [8].

The quality of the registration also depends on the quality of the object's model that is used. In some cases the model is not completely known and its unknown parameters are estimated along with the projection parameters. In any event, all registration algorithms come down to matching corresponding features between one view of an object, the 3D model of the object, and/or other views of the object. Features of the 3D model can be either geometrical features (e.g. crest lines) or texture features or patches. Features of the 2D image can be either points of interest (e.g. corners or edges of image patterns) or image patches. Naturally, 3D geometrical features match with points of interests and 3D texture patches match with image patches.

This work is supported in part by the FR RNRT SEMANTIC-3D project

In practice, matching features can be a difficult problem. This is mostly due to the fact that they cannot always be localised with enough accuracy and reliability. For example, while it is possible to define and find geometrical features in a 3D model, it is almost impossible to find matching geometrical features in a 2D view of a *textured* object (except for the apparent contour). In this case, a possibility is to match texture patches instead, but this is reliable only if the texture model approximates the real object with enough accuracy. There is a similar problem when matching two views of the same object under two different point of views (e.g. in stereovision techniques): the texture may not appear to be strictly the same due to lighting conditions that appear different from both points of view. In that case it may be useful to model the lighting differences as parameters to be estimated too.

3. Applications of synthetic/synthetic projective registration

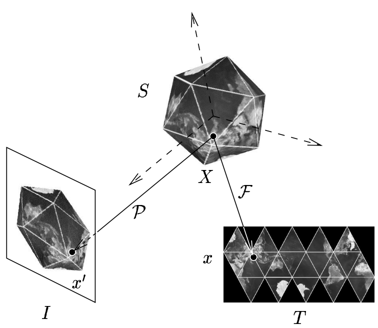

Figure 1. Texture reconstruction from a 2D view.

Registering a 3D model with a computer-generated 2D view of it can be useful to assess the efficiency of a registration algorithm that is designed to be used with images of real objects, because in a computer-generated experiment the exact solution can be known beforehand. This kind of registration can also be seen as a kind of reverse-engineering (find the parameters used to produce an image from a given 3D model). However, as mentionned in the introduction we know of only one real application requiring the registration of a 3D model with a computer-generated 2D view of it, and for which a specific algorithm should be designed: texture-based watermarking of 3D objects.

In this application, a 3D object S (cf. figure 1) has a texture image T mapped onto it via a texture mapping function $\mathcal{F}$, and is then projected in a 2D image I via a perspective projection $\mathcal{P}$, but before texture mapping, the texture image is watermarked using an adapted still-image watermarking algorithm, so that after texture mapping and perspective projection, the watermark should be present in some way in the resulting image I. Now the aim is to extract the watermark from the image I. To do this we need to first reconstruct T as well as possible from I and then extract the watermark from the recovered T. The reconstruction of T is possible only if we know $\mathcal{P}$ and $\mathcal{F}$. In order to know $\mathcal{F}$ we must assume that the original unwatermarked object is known (which implies a non-blind watermark extraction mode). And in order to know $\mathcal{P}$, we must perform a projective registration between the object S and its image I. This 3D watermarking application is presented in more details in [2]. For now it suffices to mention the conditions in which the synthetic/synthetic projective registration must take place.

First, a high accuracy is required, that is, we must estimate a projection $\mathcal{P}$ that projects the textured object S onto the image I with subpixel accuracy. Second, the object to be registered is a textured object, which implies that we should match texture patches and not geometrical features. Then, it should be possible to achieve an accurate match of texture patches because the object and its texture are known a priori, which is hardly possible when registering real world images. Finally, we should deal with synthetic illumination, which is unknown a priori.

The fact that no classical computer vision algorithm is primarily designed to work with computer-generated images and that therefore it does not take advantage of the fact that the object's texture can be perfectly known a priori, is the first motivation for developing an algorithm based on this a priori knowledge.

4. Texture-based synthetic/synthetic projective registration algorithm

4.1. Overview of the algorithm

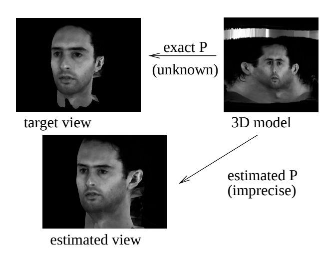

Figure 2. The registration problem: the aim is to know what projection P was used to obtain a given view (target view) of a given 3D object. If the estimated projection is not very accurate, the corresponding view of the 3D object does not perfectly match the target view.

The problem of registering a known 3D object and 2D view of it is illustrated in Fig. 2.

A possible approach consists in determining a set of matching features (e.g. points or contours) in both the 3D object and the image and then directly computing the projection that minimizes the distance of the projected 3D feature with the image features. For instance the estimation of the 11 parameters of a general perspective projection matrix is theoretically possible with the knowledge of at least 6 matching points, even though in practice using so little data leads to unstable computations and inaccurate estimates.

Another approach is to estimate the rendering parameters (projection and lighting conditions) that minimize the error (e.g. mean square error) between the target image and the image computed from the parameters being estimated [6, 7].

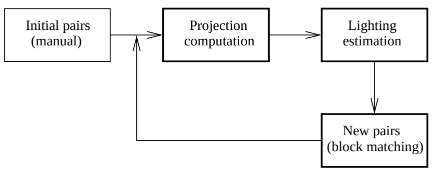

Figure 3. Overview of our projective registration algorithm.

As shown on figure Fig. 3, the proposed projective registration algorithm iteratively improves the estimated projection in the following three steps — 1) Projection estimation from pairs of matching points; initial pairs are obtained manually, and better pairs are computed at each iteration — 2) Lighting estimation; this operation aims at estimating the lighting conditions used in the target view so as to reproduce them in the approximate view before block-matching — 3) Computation of new pairs using block-matching — These steps are detailed in the following.

4.2. Computation of the projection matrix

For an in-depth exposition of the 3D computer vision theory, and of the notions used in the following, the reader is refered to existing publications such as [1].

Given a set of 3D-2D pairs $\{(X_i, x_i)_{i\in I}\}$ (expressed in homogeneous coordinates) we compute the 3×4 projection matrix P that verifies $PX_i \equiv x_i$ for all $i \in I$ (where the equivalence relation $\equiv$ means collinearity).

In our case the system is over-determined and the X_i and x_i have a limited accuracy, so that the solution can only be determined with a limited accuracy by minimizing some criterion. We used the classical method of rewriting each $PX_i \equiv x_i$ as a two-equation linear system and of solving the global linear system in the least-mean-square-error sense.

It is well known that this solution cannot be expected to minimize the reprojection error, but it can be improved in two ways: applying a gradient descent on the projection parameters towards the minimum reprojection error, or normalizing the coordinates X_i and x_i before solving the linear system as demonstrated in [3]. We chose the latter option.

4.3. Lighting estimation

In order to make the block matching possible we first have to compensate illumination differences between the original object and the 2D view of it produced with synthetic lighting. In a preliminary experiement we used a simple lighting model: the object's surface can only diffuse light (no specular reflection) and there is only ambient light with arbitrary color and intensity and one directional light source with arbitrary color, intensity and direction. More complex lighting models may have to be used if confronted with very realistic rendering, which includes several light sources, and/or very realistic surface properties [4].

The parameters of the simplified lighting model that minimize the mean squared error between the target view and the resynthesized approximate view are found by gradient descent. It must be noted that in the first iteration, the error criterion may be biased if the two views are not already well aligned. However, even if imperfect, the lighting estimation helps the block matching.

4.4. Computation of improved 3D-2D pairs

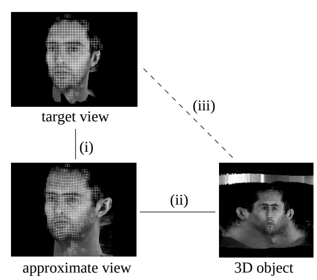

Figure 4. Improving 3D-2D pairs of points using block-matching between 2D views.

To improve the estimate of the projection matrix between the known 3D object and the target 2D view, we find 3D-2D pairs between them using block-matching between the target view and the currently estimated view. As shown in Fig. 4, the (i) pairs are obtained by block-matching, the (ii) pairs are known because the 2D points of the view computed from the current estimate of the projection are in known correspondance with the 3D points of the 3D object (via this very projection), finally new

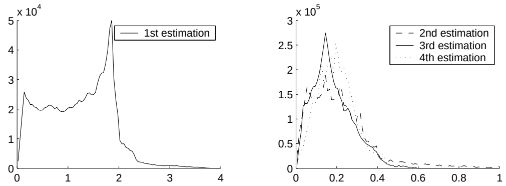

Figure 5. Results of the initial manual registration (left) and first three iterations of the registration algorithm (right). Each curve represents the number of pixels (vertical axis) affected with a given reprojection error (horizontal axis, in pixels).

pairs (iii) are computed by transitivity of (i) and (ii) from which the new, and hopefully more precise, projection estimate will be computed in the next iteration.

The quality of the pairs obtained, or of the projection estimated from them, could be improved in the following ways — 1) Using a sub-pixel block-matching algorithm — 2) Considering the block-matching problem as a stereo-vision problem and using stereo-vision techniques. In fact this is a stereo-vision problem to the extent that we try to match two views of the same object obtained from two (slightly) different points of views (target one and estimated one).

5. Experimental results

To estimate the accuracy of the projective registration, we produced a synthetic image from a known 3D model and under known conditions (lighting and point of view) in a 2D view similar to that of Fig. 2.

The algorithm was initialized by manually picking 10 pairs of corresponding points on the model and view. Fig. 5 shows the error between the actual projection and the estimated projection for the first four iterations of our algorithm. This error is computed as the distance in pixels between each pixel of the target view to the corresponding pixel in the estimated view, and is displayed as a repartition histogram. The first histogram shows the error of the projection computed from the manually selected pairs and the next ones show the results of the optimizing loop. We observe that in this experiment, the first estimation is already good enough to allow the algorithm to converge quite quickly. In fact the third estimate is the one with least mean error and we did not observe any further improvement of the projection estimate afterwards.

6. Conclusion

In this paper we presented a projective registration algorithm that is closely related to well-known computer vision problems such as camera calibration, stereo-vision, object tracking.

The main reason for not directly applying existing algorithms is that the context where we need to achieve a projective registration is quite specific. First, we only use synthetic images instead of real images, which allows us to correctly model the virtual camera as a pure perspective projection with no lens distorsion. The 3D model of the object represented in the synthetic image can also be exactly known (at least this is the assumption made in the non-blind texture-based watermarking application). All this should allow to achieve a very good registration accuracy, which is also a requirement for the application of watermarking. Finally the objects used or targeted by the application of watermarking, such as human faces, are textured objects which may not have very neat features (such as “corners” or other geometrical patterns, as usually used on calibration objects). Thus a textured-based projective registration method was also prefered.

The presented method achieved a registration accuracy of about 0.2 pixel under certain conditions: good enough initial registration, synthetic lighting consisting of ambient light and one directional light source with no specular reflection.

7. References

[1] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.

[2] E. Garcia and J.-L. Dugelay. Texture-based watermarking of 3D video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(8), August 2003.

[3] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 19(6):580–593, June 1997.

[4] S. R. Marschner, B. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In *Proc. of 11th Eurographics Workshop on Rendering*, Brno, Czech Republic, June 2000.

[5] P. J. Neugebauer and K. Klein. Texturing 3D models of real world objects from multiple unregistered photographic views. In *Proceedings of Eurographics ’99*, pages 7–11, Milan, September 1999.

[6] F. Pighin, R. Szeliski, and D. Salesin. Resynthesizing facial animation through 3D model-based tracking. In *International Conference on Computer Vision*, pages 143–150, Corfu, Greece, 1999.

[7] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3D morphable model using linear shape and texture error functions. In *Proc. of ECCV*, 2002.

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.