From Image Sequence to Virtual Reality

Jérôme Blanc, Roger Mohr

From image sequence to virtual reality

Jérôme BLANC Roger MOHR

LIFIA - INRIA
46, av. Félix Viallet
38031 GRENOBLE CEDEX 1
FRANCE

phone : +33 76 57 43 28 *fax :* +33 76 57 46 02 *e-mail :* Jerome.Blanc@imag.fr

KEY WORDS : transfer, matching, reprojection, epipolar, trilinear.

ABSTRACT :

This paper presents a way to explore a 3D scene defined by 2D views : given some 2D views of the same scene, which we call *reference views*, we want the user to be able to move a virtual camera, so as to generate any other view of the scene, and carry out a virtual visit of the scene.

We will show how we applied the trilinear relations stated by [Sha 94] to this so-called *transfer* problem. This approach avoids any kind of 3D reconstruction, thus allowing us to deal with real and complex images. We also successfully experimented on outdoor scenes, taken with a home video camera.

The algorithm consists in two steps : it starts with a dense matching between the reference views ; each matched couple is then reprojected using the trilinear relations.

This technique has numerous applications, among which virtual realities, or high rate video compression. The images obtained are more realistic than any 3D model we could have computed, while being geometrically correct.

1 Introduction

This paper presents a way to explore a 3D scene defined by 2D views : given some 2D views of the same (static) scene, we want the user to be able to move a virtual camera, so as to generate any other view of the scene, and carry out a virtual visit of the scene. This technique has numerous applications, among which virtual realities, or high rate video compression.

The so-called *transfer* problem has an obvious solution : we can use some 2D views to build a 3D model of the scene, which we reproject afterwards on the plane of the virtual camera. This approach raises two problems :

- if we want to compute a euclidean model, the cameras must be calibrated. Calibration is a delicate (and off-line) process ; one has to place a calibration grid in front of the cameras, then to compute the projection matrices. This can be an arduous process. Besides, this can't be done without any physical access to the cameras, e.g. if only a video input-flow is available, which can be the case if we apply this technique to compression.

- once the position of 3D points are computed, it's still a partly unsolved problem to build the corresponding 3D model (surfaces and curves). In fact, it's getting nearly impossible for a "real" scene, especially where there are three-dimensional textures, like for instance wrought iron. Moreover, it may be useless if we don't want to edit the structure afterwards but just display it under another point of view.

What we propose is to achieve the same goal without calibrating the cameras nor reconstructing any 3D-model. The tools used for this purpose are originated from standard bundle equations, from which the epipolar constraint and trilinearity constraints can be derived. The process consists in two steps :

1. Match the two images to recover the scene structure ; this is a necessary condition to extract some depth information from the stereoscopic pair. We briefly describe the algorithm we used in section 2.

2. For each point for which we could get this information, reproject it on the plane of the virtual camera ; we present such a method in section 3.

Finally, a quick discussion about possible applications and a conclusion will be presented in section 4.

2 Matching

To illustrate the process, we will use the "MOVI house" scene (see figure 1). The first two pictures, named `im1` and `im2`, are called *reference views*, and define the 3D scene. The third picture named `im3` is what we want to obtain : another view of the same scene, under another viewpoint. Notice that here the third viewpoint doesn't lie inbetween the reference views. Our aim is to synthesize an extrapolated image as close as possible to `im3`.

To recover the depth of a 3D point seen in 2 images, we first need to know where this point projects in the images. Knowing the depth (in a projective sense), we can then transfer the point in the third view. Note that the information we get is not euclidean, and would only allow us to obtain a *projective* model of the scene [Moh 92, Fau 92]. A real euclidean model would be impossible to compute, since the cameras are not calibrated.

2.1 Algorithm

Matching is a vast and largely explored problem. Our dense matching is currently implemented as a best match search along epipolar lines. We match the intensity signal along two conjugated epipolar lines using dynamic programming (see for instance [Oht 85, Gei 92]). The algorithm computes the best matching between the two signals, provided we defined a **matching cost** between two points of the signals (i.e. two pixels). Instead of using a raw absolute difference of the two pixel intensities as matching cost, we used a correlation measure between the neighbourhoods of the pixels. This allows to take into account a continuity constraint between adjacent epipolar lines.

In our experiments, this technique allowed to match about 80% of the initial points in a reasonable amount of time : on our 512×512 images, we found more than 200,000 matches in less than 15 minutes, on a SparcStation 10 (with no dedicated hardware). Figure 2 shows the points we could match (black points couldn't be matched). As we can see, not all points can find a match in the other image : e.g. the same walls don't have the same length in the 2 views (they're viewed under different angles), so they don't contain the same number of pixels, and some of them can't be matched.

3 Transfer

What we have now is a number of point matches between the 2 views. We will present here a method to transfer each point, i.e. to compute the projection of each point in the synthesized image.

3.1 Geometrical aspects

[Sha 94] has shown a set of trilinear relations to exist between the coordinates of the projected points in the 3 images. If we note :

$$p_1 = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad p_2 = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \quad p_3 = \begin{pmatrix} x'' \\ y'' \\ 1 \end{pmatrix}$$

then one form of the relation is for instance :

$$\left\{ \begin{array}{rcl} x''(\alpha_1 x + \alpha_2 y + \alpha_3) + x''x'(\alpha_4 x + \alpha_5 y + \alpha_6) + x'(\alpha_7 x + \alpha_8 y + \alpha_9) + \alpha_{10} x + \alpha_{11} y + \alpha_{12} & = & 0 \\ y''(\beta_1 x + \beta_2 y + \beta_3) + y''x'(\beta_4 x + \beta_5 y + \beta_6) + x'(\beta_7 x + \beta_8 y + \beta_9) + \beta_{10} x + \beta_{11} y + \beta_{12} & = & 0 \end{array} \right. \quad (1)$$

where $\alpha_i = \beta_i, i = 1...6$.

This relation is demonstrated in [Sha 94] using projective invariants. Another kind of demonstration can be found in [Har 94], who uses an explicit projective reconstruction : given the matrices of projection M_1, M_2 and M_3 in the 3 images, and the matched points p and p', projections of a physical point P in the two first images (reference views), we can algebraically compute the coordinates of P in the projective space. We then reproject P onto the third image, via M_3, and we obtain the coordinates (x'', y'') of P as a function of : the coordinates (x, y) and (x', y') of p and p', and the coefficients of the matrices M_1, M_2 and M_3. If we develop the equations, the 17 parameters of Shashua's relation appear as combinations of the coefficients of the three M_i.

Another way to recover the trilinearity constraints has been recently proposed by [Lon 95]. If $P = (X, Y, Z, T)^T$ projects in the three images via the three projection matrices $M_1 = (I_{3\times 3}\ 0)$, $M_2 = A_{3\times 4}$ and $M_3 = B_{3\times 4}$, we can write :

$$\begin{pmatrix} x - z \\ y - z \\ x' a_3^T - z' a_1^T \\ y' a_2^T - z' a_1^T \\ x'' b_3^T - z'' b_1^T \\ y'' b_2^T - z'' b_1^T \end{pmatrix} P^T = C P^T = 0 \quad (2)$$

The rank of C cannot be more than 3. As a consequence, all 4×4 determinants are vanishing. Developping the equations leads to the bilinear and trilinear constraints.

3.2 Results

In our tests, we were given the third image (to recover). We tracked points between the 3 images to compute the 17 parameters of the trilinear relation (for the house, we used 48 matched points, and a least squares minimization).

These parameters define the position of the virtual camera. Not every configuration of the parameters describe a *physical* situation ; they only define a perspective projection of the projective model of the scene onto the plane of the virtual camera.

The only case where all forms of the trilinear relations are degenerated is for points P lying on the line joining the optical centers. For these points, it is impossible to recover depth anyway.

See figure 3 the reprojected house we obtained. The "holes" in the synthesized image arise from missing points in the matching process, and from a stretching phenomenon : two adjacent pixels in the reference views can be transfered at quite different (non-adjacent) places in the third view. There are numerous ways to avoid the holes ; we can also simply filter the synthesized image (figure 4).

4 Discussion

We presented an algorithm to, given two 2D views of the same scene, build any other 2D view, while avoiding the burden of a 3D reconstruction. This work has strong connections with [Lav 94], and also with [Wer 94].

The process works on real images. The house pictures were taken from a distance of about 1.1 m with a standard video camera (Pulnix TM-6EX with a Schneider-Kreunach Xenoplan 17 mm lens ; pixels are about 10 μm wide) and digitized with a FG150 frame-grabber from Imaging Technologies.

We also successfully experimented on outdoor scenes, taken with a home video camera (interlaced video). Figure 5 and 6 shows the result obtained on such a scene. Difficulties arise from the matching process, especially on the leaves of the trees in the foreground, or on the waves of the river.

4.1 Applications

As an application, we can take some pictures of a real scene by any means, and carry out a virtual visit of the scene, as if a 3D model of the entire scene had been computed. The user could move around the virtual camera, and each image would be synthesized on demand. The matching phase is still quite long and needs to be computed off-line, but the reprojection could be done at near real-time rates (in our (non-optimized) implementation, we currently transfer more than 20,000 points per second). The images obtained are more realistic than any 3D model we could have computed.

Another application is very high-rate compression : given a video signal representing successive views of a scene, we could extract some carefully selected reference views, and then transmit only these views (themselves possibly compressed) along with the 17 parameters of the trilinear relation describing each successive view. The compression rate grows with the size of the images to transmit, because the number of parameters remains fixed. For non-static scenes, we need to compute 17 parameters by solid independently-moving object.

4.2 Open problems

- We need to take into account more than 2 reference views. The problem is :
  - What do we need to compute ? Do we need to match every point in every image ?
  - How do we transfer a point ? Do we only use its coordinates in 2 reference views ? In which views ? How do we combine more than 2 views ?
  - Which views do we need to transfer in the case of compression ? That is, which views do carry information which couldn't be found in the other views ? In which way ? To what extent ?
- Which form of the trilinear relations do we need ? Do we get more accurate results if we use simultaneously the various forms of trilinearity constraints ?

References

[Fau 92] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, May 1992.

[Gei 92] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 425–433. Springer Verlag, 1992.

[Har 94] R. Hartley. Lines and points in three views - an integrated approach. 1994.

[Lav 94] S. Laveau and O.D. Faugeras. 3-D scene representation as a collection of images. In *Proceedings of the 12th International Conference on Pattern Recognition, Jerusalem, Israel*, volume 1, pages 689–691, 1994.

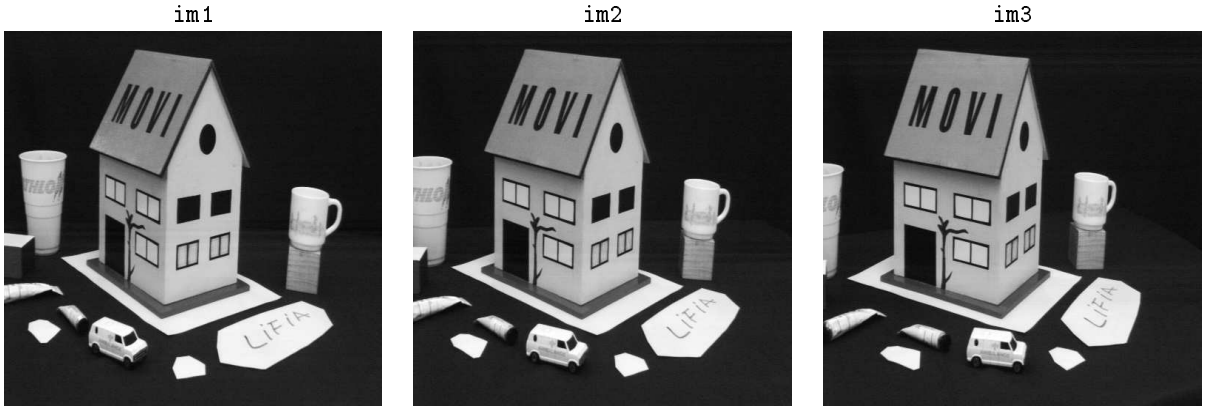
[Lon 95] Q. Long. Bilinearities and trilinearities. 1995.

[Moh 92] R. Mohr, L. Quan, F. Veillon, and B. Boufama. Relative 3D reconstruction using multiple uncalibrated images. Technical Report RT 84-I-IMAG LIFIA 12, Lifia–Irimag, 1992.

[Oht 85] Y. Ohta and T. Kanade. Stero by intra and inter-scanline search using dynamic programming. Ieee *Transactions on Pattern Analysis and Machine Intelligence*, 7(2): 139–154, 1985.

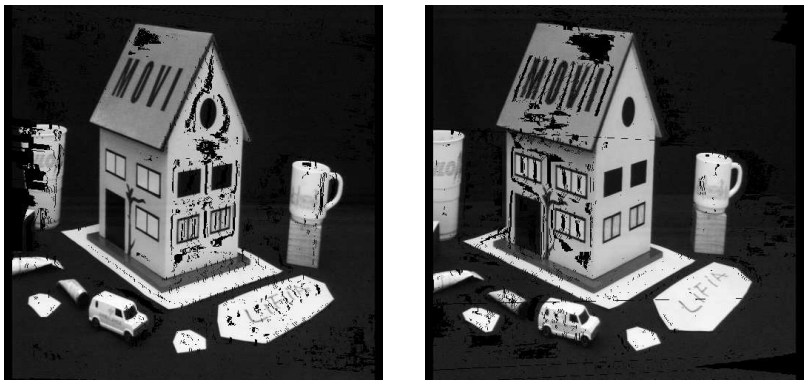
[Sha 94] A. Shashua. Trilinearity in visual recognition by alignment. In Jan-Olof Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, pages 479–484. Springer Verlag, May 1994.

[Wer 94] Th. Werner. Rendering Real-World Objects without 3-D Model. Technical report, Czech Technical University, Dept. of Control, Faculty of Electrical Engineering, Czech Technical University, Karlovo nám. 13, 12135 Praha, Czech Republic, 1994.



The 2 reference views | The image to synthesize

Figure 1: The Movi house.



Points of `im1` for which we found a correspondant in `im2`.

Points of `im2` for which we found a correspondant in `im1`.

Figure 2: The matched points ; black points couldn't be matched.

The real image. | The synthesized image.

Figure 3: Reprojection using the trilinear method.

The MOVI house.

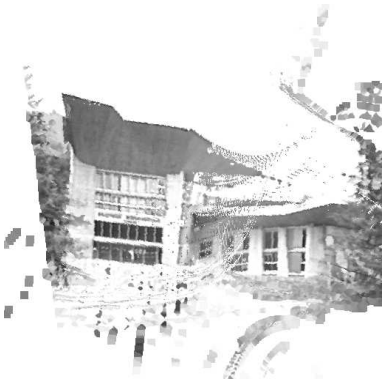
The real image.

The synthesized & filtered image.

Figure 4: Reprojection plus filtering.

The Science Library of Grenoble.

The real image.



The synthesized & filtered image.

Figure 5: Reprojection plus filtering.

The bridge.

The real image.

The synthesized & filtered image.

Figure 6: Reprojection plus filtering.