SYNTHESIS OF VIRTUAL VIEWS OF A SCENE FROM TWO OR THREE REAL VIEWS

Georges M. Quénot
LIMSI-CNRS, B.P. 133, 91403 Orsay Cedex, FRANCE
e-mail: quenot@limsi.fr

ABSTRACT

This paper presents two methods for synthesizing arbitrary views of a scene from two or three real views. Both methods rely on the extraction of a dense and accurate matching field between the images. The matching field is obtained from an optical flow computation technique based on dynamic programming. First method performs simple image interpolation with motion compensation in order to synthesize intermediate views. Transformations of the matching field allows to synthesize other views that are not simply intermediate ones. Second method is based on the reconstruction of a textured 3D surface. Both methods are demonstrated with two- and three-image sets.

1 INTRODUCTION

Synthesizing arbitrary views of a scene from two or three real views has several important applications including: dynamic animations of stererograms (3D is rendered through motion instead of static stereo), 3D movies, 3D TV and 3D teleconferencing (by synthesizing dynamically on a screen the adequate view according to the viewer head position).

The work proposed here is similar to − but does not directly extend − the work of Ernst, Izquierdo, Ohm and Karl from the Heinrich Hertz Institute [1] [2]. The approach is different both for the image matching step (optical flow [3] instead of block matching) and the image synthesis step for which two methods are proposed: enhanced image interpolation and textured 3D surface reconstruction (both allow synthesis from fully arbitrary viewpoints instead of only intermediate viewpoints between the left and right ones). Additionally, the proposed approach can take into account the information from three original views instead of only two to improve quality of the result.

2 OPTICAL FLOW COMPUTATION

Both methods presented here for the synthesis of arbitrary views require a dense (one matching vector per pixel), continuous and accurate matching field between the original views. Block matching is inadequate since it usually does not provide a continuous matching field. Instead, we use an Optical Flow computation technique based on Dynamic Programming [3] (optical flow computation consists in extracting a velocity field from an image sequence assuming that intensity or color is conserved during displacement). This technique is not described here; it has been chosen because, compared with other optical flow approaches, it has the following characteristics that makes it especially suitable for our application:

- It can be used with sequences of only two images,
- It performs a global image match by propagating continuity and regularity constraints,
- It provides a dense, coherent and continuous matching field,
- It is able to operate either on monochrome or color images.

All optical flow computation techniques, including this one, require that the displacement between consecutive images be small. For the application described here, this turns into the condition that the original images have small disparities and therefore that the viewpoints be not too far from each other (this depends on the depth and texture of the scene). In practice, the viewpoints for which views are synthesized must also be not too far from the original ones for the synthesis to remain realistic.

3 SYNTHESIS FROM ENHANCED IMAGE INTERPOLATION

3.1 Simple image interpolation

Once a dense matching is obtained from a pair of images, a simple interpolation with motion compensation directly synthesizes intermediate views without knowing anything about the camera parameters and whatever the relative camera positions are. This is a simple (but fully automatic) morphing between the two views. Such image morphing actually works even if the images are not different views of a same scene provided that they remain similar and have small disparities [4].

The optical flow computation technique used here provides a displacement vector for each pixel of the image. Additionally, it offers the possibility to compute this vector either relative to the first image (the pixel is at the begining of the displacement vector), relative to the second image (the pixel is at the end of the displacement vector) or relative to any fictitious intermediate or extrapolated image (the pixel is at a given point specified by an interpolation parameter λ of the displacement vector). In the last case, $-\lambda$ times the vector specifies the relative position of the corresponding point in the first image and $(1 - \lambda)$ times the vector specifies the relative position of the corresponding point in the second image (Figure 1). Of course, there is a single actual point to point matching between the images but, unless the displacement is globally uniform, the actual set of numerical values depends upon which image it is defined in (first, second or intermediate). Moreover, displacement vectors generally do not have integer components and, therefore, they do not really define a pixel to pixel matching.

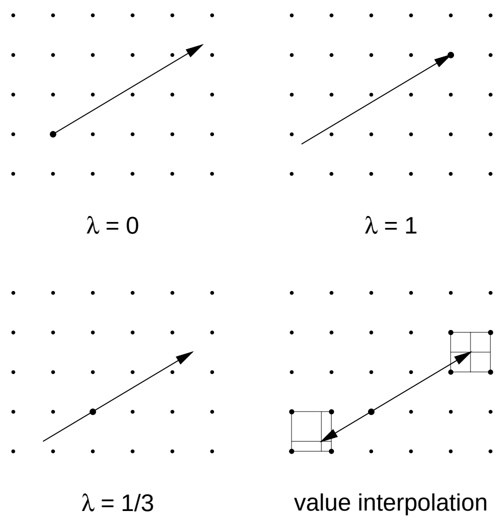

Figure 1: Motion compensated image interpolation

The fictitious intermediate image specified by an interpolation parameter λ can then be built from the displacement field defined relatively to it by assigning to each pixel the intensity (or the color) interpolated from the values associated to the corresponding points in the original images. These points generally have non integer coordinates. A bilinear interpolation is used to obtain their value (Figure 1). If a displacement leads to a point outside of the original image, the value is interpolated from the closest available image points (extrapolation).

If another optical flow computation technique is used which is able to provide the field only either in the first or the second image, or if we chose to compute the field only once for a sequence of synthesized intermediate views, it is always possible to transform a flow field defined into a fictitious image with a interpolation parameter λ_0 into another flow field defined into another fictitious image with another interpolation parameter λ_1. This is done by transporting the flow field by $(\lambda_1 - \lambda_0)$ times itself. It can be done consistently only if the flow field is dense and continuous.

3.2 Enhanced image interpolation

By simply knowing, additionally, that the pair is an horizontal stereo one, it is possible to render a motion in the vertical direction simply by adding a variable multiple of the horizontal found displacement to the vertical one. Anything else remains identical in the image synthesis process. It is still a kind of image interpolation with motion compensation. We call it "enhanced" image interpolation. The only difference is that the displacement field is modified by a simple linear transformation.

It is also possible to render a motion towards the scene with an additional hypothesis on the relief intensity or by manually setting the corresponding parameter. Again, the front motion is obtained by a simple transformation (however no longer a linear one) of the displacement field. The principle is that a pixel is moved away from the image center proportionally : to its distance from this center, to the amplitude of the front motion, and to an affine function (whose coefficients are tuned using the hypothesis on the relief intensity) of the horizontal found displacement.

Any combination of these three motions plus dilatation, rotation and pan allows to synthesize any virtual view from a stereo pair with no or very little camera information. The results are "first order approximations" of what the correct ones would be but this do not lead to noticeable artefacts if the original viewpoints are not too far one from each other.

If three views are available, it is possible to synthesize, by simple image interpolation with motion compensation, views that are intermediate between the three original ones. Horizontal and vertical motion rendering can be obtained without any camera information. The new viewpoint is simply defined by its barycentric coordinates relatively to the original ones. This could be done using a two-step two-image interpolation. However, it is possible and preferable to combine the three possible two-image matching fields into three vector fields that directly point, relatively to the current position into the fictitious interpolated image, to the corresponding points inside the three original images. Again, this is image morphing and actually works even if the images are not different views of a same scene provided that they remain similar and have small disparities. Figure 3 shows three images extracted from the Heinrich Hertz Institute "anne" sequences: left (a), right (b) and top (c) views, and the synthesized central view (d).

4 SYNTHESIS FROM TEXTURED 3D SURFACE RECONSTRUCTION

If the camera parameters are known (a simple pinhole camera model is used here), it is possible to build a textured 3D surface [4]. A 3D point can be obtained for each pair of matched points (one pair for each pixel of the fictitious image in which the displacement field is computed; we use the middle imtermediate image) as shown in Figure 2. An intensity or color value can also be obtained by interpolating the corresponding values of the matched points. It is then possible to synthesize virtual views by building facet based views of this surface (a purely emissive model is used).

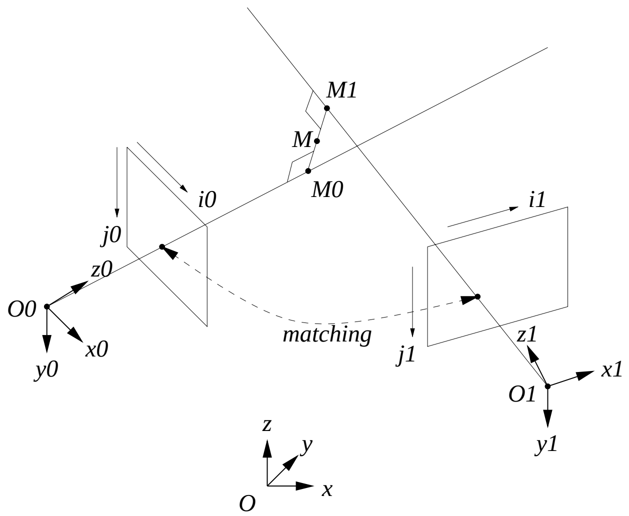

Figure 2: 3D point construction

The camera parameters need not to be very accurate and, if they are unknown, a simple model based only on focale, spacing and vergence is enough (and only one global parameter corresponding to the relief intensity has to be tuned). Even if the matching is not accurate in homogeneous regions and/or the camera parameters are approximative, the virtual views remain realistic if they are not too far from the original ones.

If three views are available, 3D points are obtained from triplets of matched points. The triplets of matched points are obtained by combining the three two-image matching fields into three vector fields that directly point, relatively to the current position into the fictitious centered image, to the corresponding points inside the three original images, exactly as this is done for the three-image interpolation. This also gives an interpolated intensity or color value for the 3D points. This improve the quality of the result compared to a reconstruction from only two original views. Figure 3 shows two synthesized views (e,f) from the three original views, synthesized side views (g) for which the viewpoints are quite far from the original ones and the corresponding 3D surfaces (h), without mapped texture, represented by slices in three orthogonal directions (spaced by 1cm in each direction).

5 CONCLUSION

Two methods for the synthesis of virtual views: enhanced image interpolation and textured 3D suface reconstruction have been described. Both rely on the extraction of a dense, continuous ans accurate matching field obtained from an optical flow computation technique based on the use of Dynamic Programming. Both methods can use two or three original views. As expected, results are better with three than with only two. Synthesis from textured 3D suface reconstruction provides slightly better results than synthesis from enhanced image interpolation.

The efficiency of the method has been demonstrated on real world images and sequences [5]. Some results, including animations, can be seen from: `ftp://ftp.limsi.fr/pub/quenot/demo/`.

A current limitation of the method is that computation of optical flow is slow and this prevents its use in real-time systems. Further investigations are carried out to increase the matching accuracy and greatly reduce the computation time using accurate camera parameters (currently, the epipolar constraint information is not used for image matching). This could also increase the acceptable disparity between the images and therefore the acceptable distance between the original and synthesized viewpoints.

References

[1] Ebroul Izquierdo and Michael Karl. Low complexity disparity estimation and object-based image synthesis for multi-viewpoint stereoscopic video conferencing. in *Intl. Workshop on Machine Vision Applications*, pages 196–199, 12-14 November 1996.

[2] Ebroul Izquierdo. Stereo matching for enhanced telepresence in 3D-videocommunications, in *IEEE Transaction on Circuits and Systems for Video Technology*, Special issue on Multimedia Technology, Systems and Applications, August, 1997.

[3] Georges. M. Quénot. Computation of optical flow using dynamic programming. In *Intl. Workshop on Machine Vision Applications*, pages 249–252, 12-14 November 1996.

[4] Georges. M. Quénot. Image matching using dynamic programming: Application to stereovision and image interpolation, in *Image Communication*, pages 265-70, Arcachon, France, 20-22 mai 1996.

[5] Georges M. Quénot. Computation of optical flow using dynamic programming and applications. In *Computer Vision and Pattern Recognition, Demo Program p. 5*, 17-19 May 1997.

(a) original left view

(b) original right view

(c) original top view

(d) synthesized center view

(e) synthesized view 1

(f) synthesized view 2

(g) synthesized side views

(h) reconstructed 3D surface side views

Figure 3: Syntyhesis of virtual views from real ones