Automatically Synthesising Virtual Viewpoints by Trinocular Image Interpolation

Stephen Pollard, Sean Hayes, Maurizio Pilu, Adele Lorusso
Digital Media Department
HP Laboratories Bristol
HPL-98-05
January, 1998

E-mail: [stp,mp,esh,lorusso]@hplb.hpl.hp.com

virtual environments, image processing, image-based rendering

We present a computationally simple and fully automatic method for creating virtual views of a scene by interpolating between three images obtained from different viewpoints. The technique is intended to give an immersive experience and a sense of viewing a real environment. The method uses correspondence points identified in the primary images. The matching of corresponding locations in the images is edge-based and relies on the automatic extraction of suitable epipolar geometries. The resulting edge correspondences are then interpolated and used in an efficient morphing algorithm that operates one scan-line at a time to perform image synthesis.

Internal Accession Date Only

1. [1]Introduction

A growing number of researchers are exploring ways of constructing static and temporally varying immersive scenes using real world image data alone.

Several approaches have been investigated. One is to capture a large number of viewpoints and use these as an environment map [3] to be applied as a texture on some imaging surface. Particular viewpoints can be generated by projecting the texture back onto the imaging plane corresponding to the user's current view. Environment maps can be obtained as panoramas composed from multiple images [1] or based upon plenoptic capture by various lens and/or mirror arrangements [8]. A number of commercial systems, including QTVR (Apple Computer), PanoramIX (IBM), IPIX (Interactive Picture Corporation) and PhotoVista (Live Picture) have been developed. Methods for morphing between environment maps have also been proposed [7]. See [12] for a good overview of image mosaic generation to capture virtual environments.

It is possible to go beyond the exploration of 2D worlds (where the viewer is constrained to a single, or predetermined set of discrete locations in the 3D environment) by recovering a dense depth map from multiple discrete viewpoints and employing a standard texture mapping technique to view that surface from an alternative viewpoint (see, e.g., [5]). Chen and Williams [2] studied the interpolation of intermediate views from 3D data. Laveau and Faugeras. [6] bypassed the reconstruction-projection paradigm, and use *transfer* with projective invariants to predict, from dense correspondences, where pixels end up in virtual projectively distorted images. A more recent approach is to directly represent the light field in the vicinity of an object [4]. Besides requiring hard-to-obtain dense correspondences, occlusion has proved to be a difficult problem for these approaches to deal with.

An approach part way between the 2D and 3D paradigms is that proposed by Seitz and Dyer [10], which uses *image morphing* techniques to synthesise viewpoints between two original images. They observe that under fairly general assumptions, veridical virtual viewpoints can be constructed by linearly interpolating corresponding uniform regions from the two images.

The work presented in this report builds upon the approach of Seitz and Dyer. First, we use a different matching and morphing strategy that does not require the images to be pre-warped and then post-warped for each viewpoint, leading to considerably more efficient rendering. Secondly, and more importantly, we extend the method to three images, thereby allowing the user to explore the change of viewpoint not only side by side but also up and down, which considerably enhances the feeling of *3D-ness* in the scene being viewed.

This approach gives surprisingly high quality and compelling virtual viewpoint reconstruction over the region between the original camera positions provided that the difference in their camera geometry is not too dramatic. The method leads naturally to the development of an immersive video media type in which multiple video tracks are augmented with a morphing track to allow the viewer to alter their viewpoint with respect to the video sequence in real time.

[1] This work has been submitted to the IEEE Computer Vision and Pattern Recognition Conference 98 for publication.

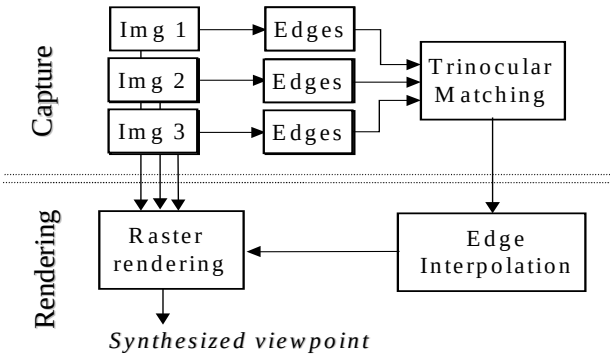

Figure 1. Outline of the trinocular view interpolation approach presented in this report (see text for details).

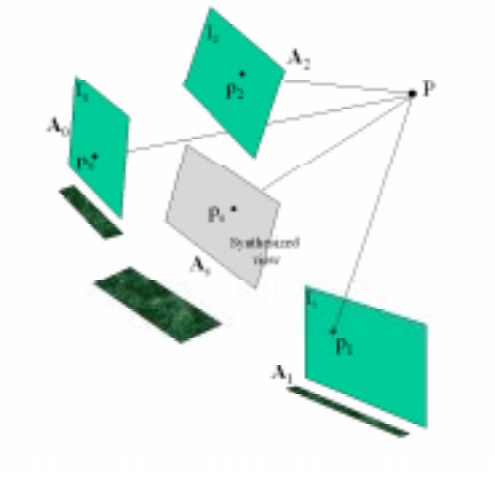

Figure 2. Geometry of the trinocular affine camera setup.

2. Overview of the Approach

The architecture for image synthesis is shown in Figure 1.

Starting from three images of a scene, arranged roughly at the corners of an triangle, we extract and match edges using a trinocular edge matching technique based upon dynamic programming and edge string ambiguity resolution.

Based upon the geometry outlined in Section 3, virtual edge images can be synthesised for any viewpoint that lies between the three cameras by linear interpolation of the three edge images based upon these edge matches.

Virtual views are rendered by a raster-based texturing technique that uses the edge matching information, to resample colour information from corresponding uniform segments of the original images and blend them into the raster segments of the interpolated image. The raster-based rendering algorithm gives the impression of surfaces sliding behind one another at occlusions and is reasonably robust to missing edge data.

The approach does not require camera calibration, as sufficient information for epipolar edge matching can be recovered from the images themselves. In addition, as opposed to full 3D representations, the sparse edge data has only a low overhead over the original images.

In the next sections, we explain the simple geometry that supports the approach and detail the techniques used to both capture (extract matching information) and render.

3. The geometry of the three-view interpolation

In this section we outline a simple physical interpretation of what interpolating in the image plane means. Note that the arguments are valid for N>=2 cameras.

Let us assume that the cameras can be approximated with an affine model (see, e.g., [11]). Let us refer to Figure 2 and consider a point $\mathbf{P}=[X,Y,Z,1]$ in the space imaged by three affine, uncalibrated cameras defined by affine projection matrices $\mathbf{A}_0$, $\mathbf{A}_1$ and $\mathbf{A}_2$ scaled such that $\mathbf{A}_{i(3,4)}=1$ (i=0..2). The projections of a point $\mathbf{P}$ onto the image planes of the cameras are given by $\mathbf{p}_0=[x_0\ y_0\ 1]^T=\mathbf{A}_0\mathbf{P}$, $\mathbf{p}_1=[x_1\ y_1\ 1]^T=\mathbf{A}_1\mathbf{P}$ and $\mathbf{p}_2=[x_2\ y_2\ 1]^T=\mathbf{A}_2\mathbf{P}$.

Let the interpolation of these three points in image plane coordinates be given by:

$$\mathbf{p}_S = (1-\beta)((1-\alpha)\mathbf{p}_0 + \alpha\mathbf{p}_1) + \beta\mathbf{p}_2 = (1-\beta)((1-\alpha)\mathbf{A}_0\mathbf{P} + \alpha\mathbf{A}_1\mathbf{P}) + \beta\mathbf{A}_2\mathbf{P} = \mathbf{A}_s\mathbf{P}$$

where $\mathbf{A}_s = (1-\beta)((1-\alpha)\mathbf{A}_0 + \alpha\mathbf{A}_1) + \beta\mathbf{A}_2$. Thus interpolation in the image plane produces the same effect as having an another affine camera $\mathbf{A}_S$.

But what does this interpolated affine matrix look like? Ullman and Basri [13] show the conditions under which linearly interpolating orthographic views produces other veridical views. Chen and Williams [2] used interpolation between range data to synthesise intermediate views and again found that valid views are produced only under some circumstances. More recently Seitz and Dyer [10] demonstrated that interpolating parallel camera images always produces valid in-between views.

Since, as we shall see later, our approach does not explicitly use parallel camera rectification as in [10], we have to understand what an interpolated affine camera matrix represents.

An affine transformation can be seen as a parametric mapping from $\Re^3 \rightarrow \Re^2$

$$\mathbf{A}_i = \mathbf{A}_i(\mathbf{v}) = \mathbf{A}_i(\theta, \vartheta, \psi, t_x, t_y, t_z, S, S_x, S_y)$$

function of the camera reference frame orientation and position, plus a shearing and two scaling components, respectively.

Now, since the tranformation is linear in translation, scaling and shearing, if no rotation between the cameras $\mathbf{A}_0$, $\mathbf{A}_1$ and $\mathbf{A}_2$ is involved, $\mathbf{A}_s$ represents a perfectly valid new viewpoint $\mathbf{V}$.

On the other hand, when rotation is involved this is no longer true. However, provided the relative rotations cameras between the cameras are small, there is a near-linear relationship between changes in the elements of the affine matrices and changes in the gaze angles. Hence, under these conditions in general we can write:

$$\mathbf{A}_s \approx \mathbf{A}_0((1-f_\beta(\beta))((1-f_\alpha(\alpha))\mathbf{v}_0 + f_\alpha(\alpha)\mathbf{v}_1) + f_\beta(\beta)\mathbf{v}_2)$$

where $f_\alpha(\alpha)$ $f_\beta(\beta)$ are non-linear functions of α and β. Thus the synthesised viewpoint, neglecting second order effects, simulates the camera being on the hyper-plane through $\mathbf{v}_0$, $\mathbf{v}_1$ and $\mathbf{v}_2$.

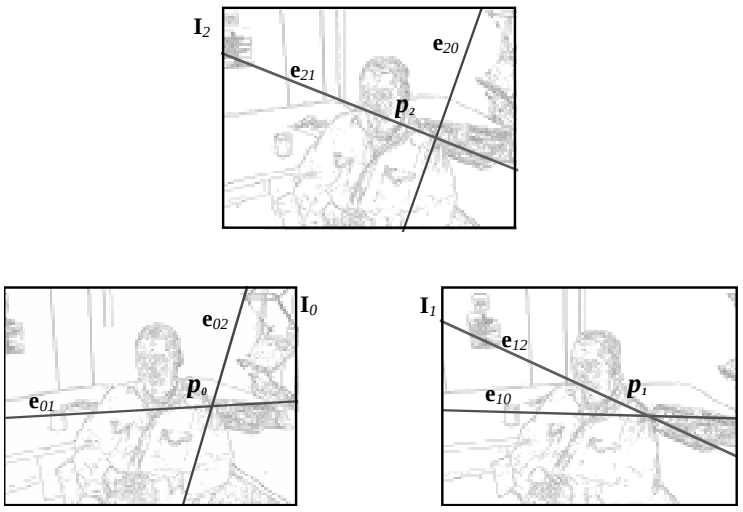

Figure 3. The trinocular matching constraint used to extend Ohta and Kanade's binocular edge matching method [9] to the three-image case. ($\mathbf{e}_{01}$, $\mathbf{e}_{10}$), ($\mathbf{e}_{21}$, $\mathbf{e}_{12}$) and ($\mathbf{e}_{02}$, $\mathbf{e}_{20}$) are conjugates epipolar lines.

4. Trinocular Matching of Edges

In this section we describe the method that has been used to match edges of trinocular triplets of images which draws from the work of [9] and extends it to the uncalibrated trinocular case. The method is composed of three parts: *I)* estimation of the epipolar geometry for each pair of cameras, *II)* trinocular matching of *edgels* (edge pixels) and *III)* matching connected strings of edgels.

4.1 Epipolar geometry estimation

The epipolar geometry for each image pair is estimated using the method proposed by Zhang *et al.* [16]. First corners are extracted in each image and a set of initial matches recovered using a local matching strength and a global relaxation process. These matches are used in turn to fit the epipolar geometry equation using the *RANSAC* (robust statistics) method.
Although a perspective version of the fundamental matrix that relates the epipolar geometries of the cameras could be used we prefer the more stable affine fundamental matrix that can be reliably fitted without an iterative non-linear estimation method.

4.2 Matching edgels

Given the epipolar geometries that relate the three images we can exploit trinocular consistency [15] as illustrated in Figure 3. Matching image points are constrained to lie on corresponding pairs of epipolar lines between each pair of views, hence correctly matched points must be mutually consistent.
Taking inspiration from the stereo matching work of Ohta and Kanade [9], the method we employed works as follows (see [17] for details).

First the edges are extracted in all images via an implementation of the Canny edge detector. Successively, a modified version of the dynamic programming (DP) method is used to match up edgels along each pair of epipolar lines of image I_0 and I_1 as in [9]. In order to extend the method to three images, and exploit the additional trinocular information, we use local edge and/or intensity properties (such as the contrast and its sign, edgel strength and orientation) of the *three* edgels $\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_2$ to compute the cost function used to build up the path of the DP table. In this way the original binocular method is naturally extended to the trinocular case. It should be noted that the DP method described produces matches between I_0 and I_1, which are in turn used to infer the match in I_2 by epipolar intersection.

4.3 Matching edge strings

The individually matched pixels that exist for a subset of edge points are processed to recover matched edge strings. This process, besides creating edge matches, helps resolve and rectify ambiguities and mismatches in the results of the initial epipolar matching that would cause ghosting when rendering (an undesirable effect also remarked in [10]).
The algorithm, which is fully described in [17], exploits edge string coherence to resolve residual ambiguity. The iterative algorithm employed is briefly outlined as follows:

While edge strings remain do:

1. **Identify the best global edge string matches for each edge string in image 1 and image 2.**
2. **Rank edge string matches in ascending order of the number of edge points matched.**
3. **Select the best edge string match**
4. **Select matches that are consistent with the edge string match and eliminate matches that are inconsistent marking all edge strings touched**
5. **Recompute best edge string match for all marked edge strings**

Upon completion matched edgel strings are combined if an underlying edgel string connects them in one or other image and the disparity measured between their endpoints satisfies a similarity constraint. Matched edge strings are extended a few pixels at each end by linear extrapolation of the underlying edge data in each image to overcome fragmentation in the edge detection and matching process. Additionally, for completeness of the synthesised images, extra edge connectivities are hypothesised between the matched edges at the left and right hand sides of the matched edge data

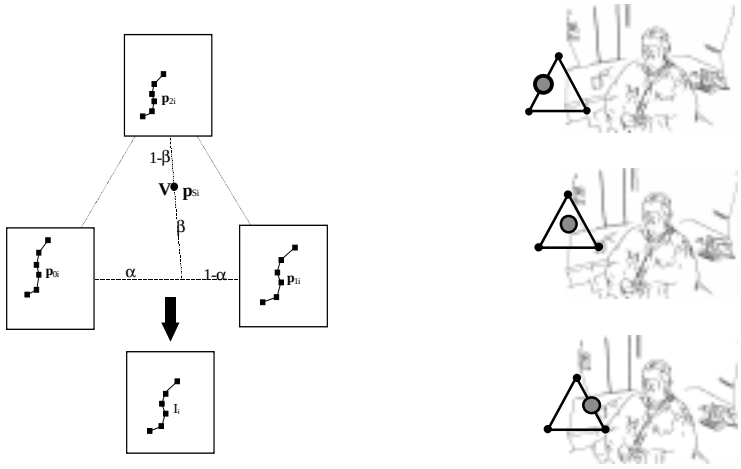

Figure 4. Illustration of the trinocular linear interpolation of edges (left) and three examples of interpolated sketches (right), which will later be filled with texture data raster-wise

5. Creating virtual views

A virtual view is created by first interpolating edgel-wise the three edge images into a virtual "sketch" and then by colouring the segments raster by raster.

5.1 Interpolating edge sketches

Rendering is based upon the linear interpolation of matched edgels, which form a virtual line sketch of the new view.

Figure 4 (left) depicts how matched edgels are interpolated to generate virtual viewpoints within the original image triple. Each string of matched edgels is interpolated according to the parameter pair (α, β) that specify the location **V** of the new view with respect to the original set. Physically α specifies a view between I_0 and I_1 and β specifies the location between that point and the location in the third image I_2. Thus, the i^{th} edge point along the string has projection into the three views at $\mathbf{p}_{0i}$, $\mathbf{p}_{1i}$ and $\mathbf{p}_{2i}$ and into the synthesised view at location $\mathbf{p}_{Si}$ given by

$$\mathbf{p}_{Si} = (1-\beta)((1-\alpha)\mathbf{p}_{0i}+\alpha\mathbf{p}_{1i}) + \beta\mathbf{p}_{2i}.$$

Figure 4 (right) gives three real examples of interpolated sketches with an indication of where they stand in the virtual viewing triangle range.

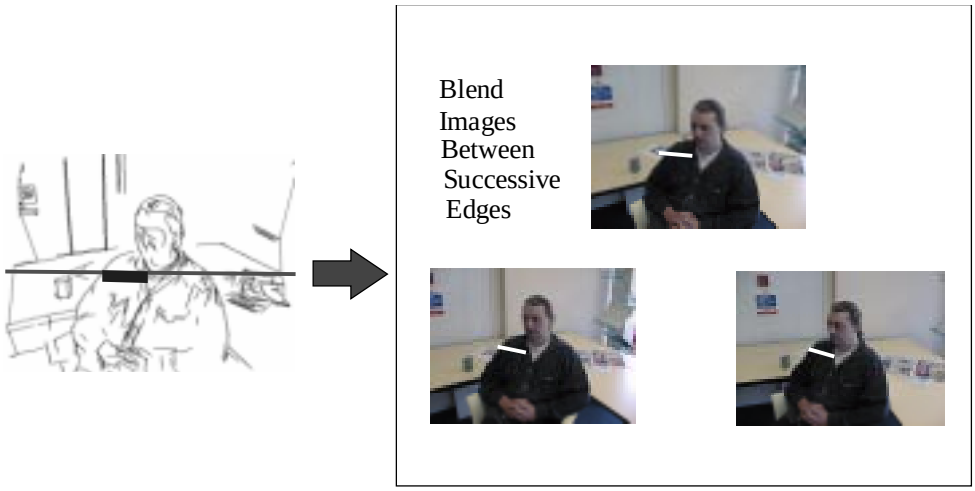

Figure 5. Raster rendering. From the interpolated sketch, the segments between intersections of rasters with edges are found (left, thick line) and corresponding the texture is fetched from the primary images (right, white lines).

5.2 Raster rendering

The interval between each successive pair of edge intersections within a raster in the virtual-viewpoint sketch is filled using a combination of the image data obtained from corresponding intervals in the primary images. A similar method was adopted by Seitz and Dyer [10] for binocular images obtained from parallel camera geometries (obtained from more general viewing geometries by image re-projection/rectification) but it is not straightforward however to extend their approach to situations involving more than two cameras.

Figure 5 shows, on the left, a selected raster within a virtual viewpoint of which the section between a pair of successive interpolated edges has been marked. On the right the corresponding intervals in the primary images have also been marked.

The algorithm uses an *intersection table* to efficiently identify the projection of the raster interval with respect to the 3 original views. This table is built incrementally during edge interpolation stage (Section 5.1). Each entry consists of an edge intersection with respect to a raster of the virtual view and the co-ordinates of corresponding points in each of the three views. The table is indexed spatially, ordered along the raster, so that intervals are efficiently obtained from successive entries.

The rendered pixels in the raster interval are thus obtained by blending the pixels from the three corresponding intervals in the primary images. As with standard image morphing techniques [14] the blend of the pixel contributions from each of the three images is linearly weighted according to the viewpoint parameter pair (α, β). Aliasing artefacts are reduced by using pixel-level bilinear interpolation when sampling the primary images [14].

Figure 6. Snapshots from the renderer. All images are synthesized viewpoints and the three vertices, in particular, are extrapolated beyond the three primary images.


Figure 7. Undesirable edge fragmentation while interpolating beyond range (left) and correction by depth of field ordering (right).

6. Violating Monotonicity

The preservation of monotonicity, cited by Seitz and Dyer [10] as an assumption for the applicability of their method also limits the scope of this approach and prevents extrapolation of the viewpoint beyond the limits of the original images. In practice monotonicity comes down to a constraint on the order of edge intersections in each view. Within the limits, edges visible in all three views which have the same order with respect to the epipolar geometry in each are guaranteed to interpolate without violating order. However, this is not generally true, if we extrapolate our viewpoint outside the region within the original image triple (i.e. if we violate either of the conditions $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$). Given that the linear image interpolation constraint may be valid for some way beyond the original image viewpoints a rendering scheme that does not rely on monotonicity proves to be desirable.

In order to overcome these problems, we have developed a raster rendering heuristic that analyses all the edges that intersect a single raster. They are ordered in terms of depth (back to front) based upon a disparity metric. Each edge then renders up to the furthest along the raster edge intersection whose projection into the three views satisfies order and for which the line joining the projection of the two points is free from occluding edges.

Figure 7 shows two versions of the rendered image from a viewpoint given by (α=1.5, β=0). This results in order violations of the interpolated edges; particularly noticeable between the head and features from the background wall. For the image on the left which uses the straight-forward rendering technique order violation results in unsightly edge fragmentation while the corrected image on the right is free from distortion. In both cases values of α and β used for intensity blending are clamped at 0 and 1 as appropriate. Figure 6 shows synthesised images for a range of viewpoints within and beyond the three primary images. While a number of minor image defects are visible (due mainly to difficult occlusions in the scene) the result gives an effective and compelling illusion of images captured from intermediate world locations.

The results are hard to appreciate just from snapshots, however the effect while “navigating” the view range is compelling, giving a true sense of three dimensionality, both it terms of object deformation and motion parallax.

7. Implementation issues

The method presented in this report has several applications but, in common with related works, it has a heavyasymmetry between the slower capture phase (edge detection and matching) and the near-real time raster rendering.

The current non-optimised implementation of the capture part runs at about 30 second per frame on an HP-9000™ workstation, but at least one or two orders of magnitude speed improvements could be achieved though HW or DSP-assisted edge detection and by employing integer arithmetic in the edge matching phase. The epipolar geometry estimation need, of course, be performed only once, provided that the cameras are not moving.

On the other hand, rendering can be achieved under Windows™ on a 200MHz Pentium™ PC at a rate of over ten 640x480 frames a second by simplifying the rendering algorithm to warp only the single closest image to the current viewpoint and exploiting nearest-pixel image sampling (rather than using bi-linear interpolation).

8. Discussion

Image-based rendering has in the past few years become a viable alternative to range data in applications such as photo-realistic immersive environments, entertainment and even graphics HW architectures.

In this context, this report presents a method for the automatic reconstruction of images at virtual viewpoints using linear interpolation between uncalibrated image triples. The method has a number of advantages with respect to methods that have recently been investigated (e.g. [11,10]), notably it uses a sketch interpolation and colouring technique that makes it possible to extended the method to three images and achieve fast rendering, a key factor if special HW is not available.

Because of the well-known problem with getting dense (even at edgel level as we did) correspondences, the method works best if the baseline between the cameras is small with respect to the viewing distance, as this improves the quality of stereo matching and reduces the number and degree of occlusions. Despite this, we have experimented with a depth-ordered rendering technique that allows to correct for many artefacts and to extrapolate beyond range to enhance the viewing experience.

The method has also been successfully applied to a number of short video sequences of animated objects (people). Each triplets of corresponding frames is processed afresh to recover matched edge strings. The whole sequence can then be played and the viewpoint with respect to it changed in real time.

We are currently investigating better edge matching strategies, in particular temporal consistency constraints (edge tracking) that could be used to improve quality and matching speed, and ways of smoothly switching between image triples to achieve an extended navigable area.

References

1. Chen, S.E., "QuickTime® VR- An Image-Based Approach to Virtual Environment Navigation", *Proc. SIGGRAPH 95*. In *Computer Graphics*, pp29-38, 1995.
2. Chen, S.E. and Williams, L. "View interpolation for image synthesis", , *Proc. SIGGRAPH 95*. In *Computer Graphics*, pp279-288, 1993.
3. Greene, N., "Environment Mapping and Other Applications of Word Projections", *IEEE Computer Graphics and Applications*, Vol 6, No 11, pp 21-29, 1986.
4. Gortler, S.J, Grzeszczuk, Szeliski, R. & Cohen, M.F., "The Lumigraph", *SIGGRAPH 96, In Computer Graphics*, pp 31-42, 1996.
5. Kanade, T., Narayanan, P.J. & Rander, P.W., "Virtualised Reality: Concepts and Early Results", *Proc. IEEE Workshop on Representation of Visual Scenes*, pp 69-76, 1995.
6. Laveau, S & Faugeras, O., "3D Scene Representation as a Collection of Images and Fundamental Matrices", INRIA Tech Report 2205, February 1994.
7. McMillan, L. & Bishop, G., "Plenoptic Modelling". *Proc. SIGGRAPH 95, In Computer Graphics*, pp 39-46, 1995
8. Nayer, S.K., "Catapiopetric Omnidirectional Camera", *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pp 482-488, 1997
9. Ohta, Y. & Kanade, T., "Stereo by Intra- and Inter-Scanline Search", *IEEE Trans. PAMI*, Vol. 7, No. 2, pp139-154, 1985..
10. Seitz, S.M. & Dyer, C.R., "Physically-valid view synthesis by image interpolation", *In Proc. IEEE Workshop on Representation of Visual Scenes*, pp 18-25, 1995
11. Shapiro, L.S., *Affine Analysis of Image Sequences*, Cambridge University Press, 1995.
12. Szeliski, R., "Video Mosaics for Virtual Environments", *IEEE Computer Graphics and Applications*, 22-30, March 1996.
13. Ullman, S. & Basri, R., "Recognition by Linear Combinations of Models", *IEEE Trans. PAMI*, Vol 13, No 10, pp 992-1006, 1991.
14. Wolberg, G., *Digital Image Warping*, IEEE Computer Society Press, 1990.
15. Yachida, M., "3D Data Acquisition by Multiple Views", *Third International Symposium of Robotics Research*, Faugeras, O.D. & Girault, G. Eds, MIT Press, 1986.
16. Zhang, Z. & Deriche, R., "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry", INRIA Tech. Rep. 2273, 1994.
17. Pollard, S, Hayes, S., Pilu, M., Lorusso, A., "Method for automatically synthesising virtual viewpoint by image interpolation", HPL Technical Report, *In Preparation*