View Synthesis by Trinocular Edge Matching and Transfer

S. Pollard , M. Pilu, S. Hayes and A. Lorusso
Hewlett-Packard Laboratories
Bristol (UK) BS12 6QZ
[stp|mp|esh|lorusso]@hplb.hpl.hp.com

Abstract

This paper presents a novel automatic method for view synthesis (or image transfer) from a triplet of uncalibrated images based on trinocular edge matching followed by transfer by interpolation, occlusion detection and correction and finally rendering. The edge-based technique proposed here is of general practical relevance because it overcomes most of the problems encountered in other approaches that either rely upon dense correspondence, work in projective space or need explicit camera calibration. Applications range from immersive media and teleconferencing, image interpolation for fast rendering and compression.

1 Introduction

A number of researchers have explored ways of constructing static and temporally varying immersive scenes using real world image data alone. Initial efforts include capturing a large number of viewpoints and use these as an environment map [6] to be applied as a texture on some imaging surface.

In this paper we are interested in actually generating new views from a small set of existing ones. This problem, called *view synthesis*, has received considerable recent interest [10],[12],[5],[9],[1],[4],[3],[7],[14] because, although in its simplest form it is better replaced by a short video sequence (a spatially dense version of [6]), its potential applications are much more far reaching. For instance we have experimented with applying the method to image sequences in order to obtain a video whose viewpoint can be changed arbitrarily within a certain range. View synthesis is also an excellent compression technique, and the one-view point version, *frame interpolation*, has been used for frame-rate conversion and frame replacement.

Researchers have investigated several approaches to solve this view synthesis problem, which can be categorised as reconstruction-projection, projective transfer and forms of image interpolation/morphing.

In the first category there is the work of Kanade et al [8] in which dense 3D surfaces are recovered from multiple calibrated viewpoints and standard texture mapping is employed to view that surface from alternative views. Chen and Williams [5] have also studied the interpolation of intermediate views in the context of 3D graphics using the

BMVC 1998 doi:10.5244/C.12.77

perfect dense correspondences implicit in the synthetic data. In the second category is the work of Laveau and Faugeras. [9] and more recently Avidan and Shashua [1], which use *projective transfer* to predict, from dense correspondences, where pixels end up in virtual projectively distorted images[1].

Besides requiring hard-to-obtain dense correspondences, approaches in either of the first two categories suffer from occlusion artefacts at depth discontinuities even under modest changes in viewing angle (overcome in [8] by integrating a large number of viewpoints).

In the third category simple image interpolation and intensity blending are used to morph novel views between an original set. This approach was inspired by the work of Ulman and Basri [15] in the area of object recognition. An early example [2] uses dense correspondences recovered by optic flow methods to morph face pose and expression. More recently Seitz and Dyer [12] have used *image morphing* techniques to synthesise viewpoints between a pair of images. They have shown that the generated viewpoints are physically valid if the two images are first re-projected to conform to a parallel camera geometry. This approach has the advantage that matching and rendering can be performed independently on each raster of the parallel camera image pair. This leads to a simplified rendering strategy that simply interpolates intensity information on the basis of matched edges along the rasters. Between the edges, intensities will vary only slowly and so the details of their interpolation will not greatly affect the overall impression.

The present work proposes a new rendering strategy for view synthesis that draws upon [12] but that uses an edge-based representation as intermediate step. In this scheme new views are generated by first creating a virtual sketch by transferring matched edges strings into the virtual view and then proceeds to colour the uniform regions between matched edges points along each raster of the virtual view.

It is important to note that the method with which one performs the transfer of the edge strings into the image associated with the virtual viewpoint is irrelevant. Once full edge correspondence has been established, one could use calibrated cameras and perform Euclidean reconstruction and projection, the physically correct projective transfer as in [1], interpolate from rectified images such as [12] or even use “approximate” simple linear interpolation as suggested by [15].

This approach has the considerable advantage that it does not rely on dense correspondence and yet can be applied generally to generate new viewpoints and not just those along the inter-ocular axis connecting the optical centres of the two cameras, as is the case of [12]. Specifically, the method described in this paper uses linear interpolation among three images and allows view synthesis as if the observer moves freely in two dimensions (up & down and left & right).

As with [12], rendering by morphing the intervals between edges along the rasters has the desirable property that depth discontinuities do not result in gaps or localised distortions as is the case with methods that depend upon dense correspondences (constructed from a small number of view points). More importantly we also show how to deal with the situation where the order of edges in the virtual view is not preserved. This allows us to render from view points outside the region covered by the initial views which would otherwise require dense correspondence.

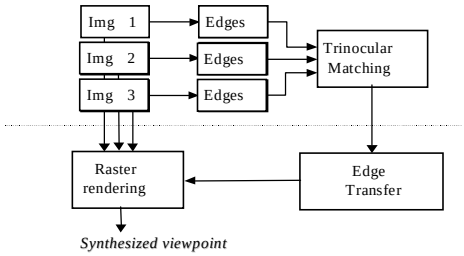
[1] Euclidian geometry and hence transfer can in fact be derived directly from 3 or more un-calibrated cameras.

The method we show here is fully automatic and does not need any manual intervention. We have applied it to complex real life scenes and the compelling results, albeit showing some artefacts due to occasional edge mismatches, indicate the definite validity of the method.

2 Overview of the Approach

This section provides an overview of the approach, which is outlined below.

Starting from three images of a scene, arranged roughly at the corners of a triangle, we extract and match edges using a trinocular edge matching technique based upon dynamic programming and edge string ambiguity resolution and using epipolar geometry extracted automatically from the images. Again, the approach does not require camera calibration, as sufficient information for epipolar edge matching can be recovered from the images themselves.



A virtual edge (sketch) image can now be synthesised for any viewpoint that lies between the three cameras by transfer of the three edge images based upon these edge matches (Section 4.1).

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.

3 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 loosely from the work of [11] 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.

EPIPOLAR GEOMETRY ESTIMATION. The epipolar geometry for each image pair is estimated using the method proposed by Zhang *et al.* [18]. 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 have been used we have preferred the more stable affine fundamental matrix [13] that can be reliably fitted without an iterative non-linear estimation method.

MATCHING EDGELS. Given the epipolar geometries that relate the three images we can exploit trinocular consistency [17] as illustrated in Figure 1. 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. The method for stereo matching works as follows.

First the edges are extracted in all images via an implementation of the Canny edge

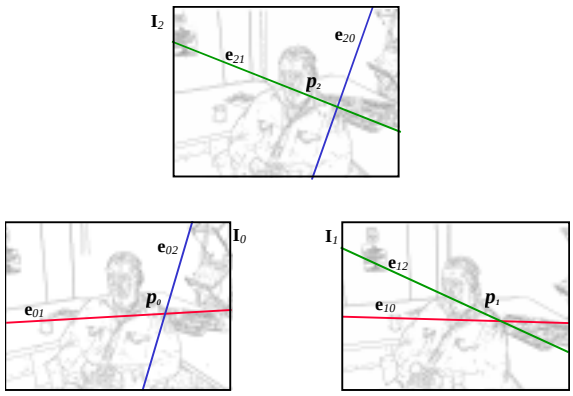


Figure 1. The trinocular matching constraint used to extend Ohta and Kanade's binocular edge matching method [11] to the three-image case. ($\mathbf{e}_{01}$, $\mathbf{e}_{10}$), ($\mathbf{e}_{21}$, $\mathbf{e}_{12}$) and ($\mathbf{e}_{02}$, $\mathbf{e}_{20}$) are conjugates epipolar lines.

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 [11]. 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.

MATCHING EDGEL 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 [12]). The iterative algorithm employed exploits edge string coherence to resolve residual ambiguity is briefly outlined in the box. 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

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**

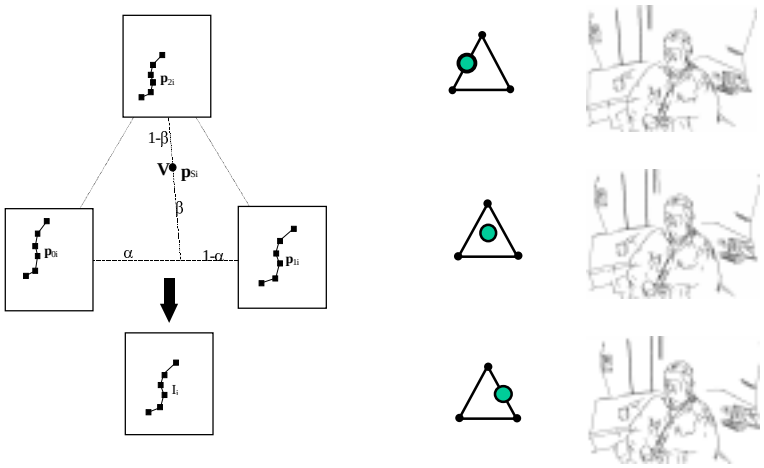


Figure 2. Illustration of the trinocul ar linear interpolation of edges (left) and three examples of interpolated sketches (right), which will later be filled with texture data raster-wise

connectivities are hypothesised between the matched edges at the extreme left and right hand sides of the matched edge data .

4 Creating virtual views

A virtual view is created by first transferring edges into a virtual "sketch" and then by colouring raster by raster segments between edges.

4.1 Virtual edge sketches by interpolation

As we explained in the introduction, the main contribution of the method is that we base the synthesised view rendering on an edge image generated as if it was taken from another, virtual viewpoint.

This can be done in several ways. The most correct and elegant way of doing it is by employing transfer of all matched edge points in Euclidean or projective/tensor space as in [1]or [9] or by linear interpolation of edges in the rectified images as in [12]. The quest for perfect image transfer out of the scope of this work so we have instead used simple linear interpolation in image space as in practice we note that more mathematically refined methods only marginally affect the final results for the small image displacements we employ. Note that in [1] and [9] the assumption of small displacements is implicit in the fact that they need dense correspondence that are very difficult to recover otherwise.

Let us now assume that the three cameras can be approximated with an affine model and consider a point $\mathbf{P}=[X,Y,Z,1]$ in the space imaged by three affine, uncalibrated cameras defined by the 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 another affine camera $\mathbf{A}_s$.

Ullman and Basri [15] show the conditions under which linearly interpolating orthographic views produces other veridical views. More recently Seitz and Dyer [12] have demonstrated that interpolating parallel camera images always produces valid in-between views; in this work the counter examples of physical invalidity of the linear interpolation show cameras with large rotations with respect to each other where interpolation breaks down. For more realistic camera geometries this is not the case. The appendix shows that in our case with small camera displacements, $\mathbf{A}_s$ does correspond to a physically valid virtual view of the scene, as suggested by the experiments.

Figure 2 (left) depicts how matched edge points 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 $\mathbf{V}$ 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 2 (right) gives three real examples of interpolated sketches with an indication of where they stand in the virtual viewing triangle range.

4.2 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 [12] 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 3 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 three original views. This table is built incrementally during the edge interpolation stage (Section 4.1). 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 [16], 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 [16].

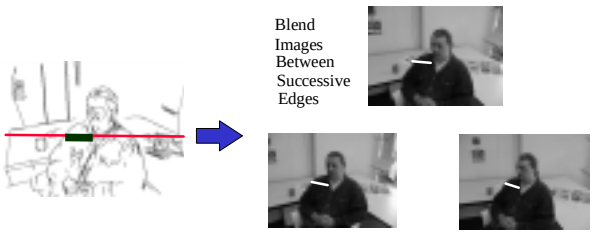


Figure 3. Raster rendering. From the interpolated sketch, the segments between intersections of rasters with edges are found (left, thick line) and corresponding the texture is fetched from the primary images (right, white lines).

5 Occlusions and Interpolating Beyond Range

So far we have assumed that the order of edges in the virtual view is preserved in each of the three original views. This is an aspect of the monotonicity constraint, cited by Seitz and Dyer [12] as an assumption for the applicability of their method. This limits the scope of the approach as it prevents extrapolation of the viewpoint beyond the limits of the original images. In order to overcome this it would seem necessary to have dense correspondences (or require dense correspondences to be inferred from the sparse edge data).

Consider Figure 4 (diagram): the part occluded side of the edge will disappear from view from one image to the other. However edges visible in all three views that have the same order with respect to the epipolar geometry in each are guaranteed to interpolate without violating order and no artefacts are produced. The result is a natural blend between the segments which through cross-dissolve gives a strong sensation of surfaces sliding one over another.

However the situation changes when we try to extrapolate edges over the limits triple (i.e. if we violate either of the conditions $0 \leq \alpha \leq 1$ and $0 \leq \beta \leq 1$), as exemplified in the leftmost example in the diagram of Figure 4. In this cases for some edges (not necessarily occluding ones) the monotonicity is no longer guaranteed and a number of artefacts crop up, as evident in Figure 4 (top-right). Given that the interpolation 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 each raster of the virtual image. They are first ordered in terms of depth, based upon a disparity metric (only the sided-ness of disparity need be given as input). Rendering then proceeds from individual edges with greatest depth (e.g from back to front). Rather than render from that edge up to the very next edge along the raster as described previously, we search along the raster for a preferred edge intersection. The best edge to render up to is in fact the edge furthest along the

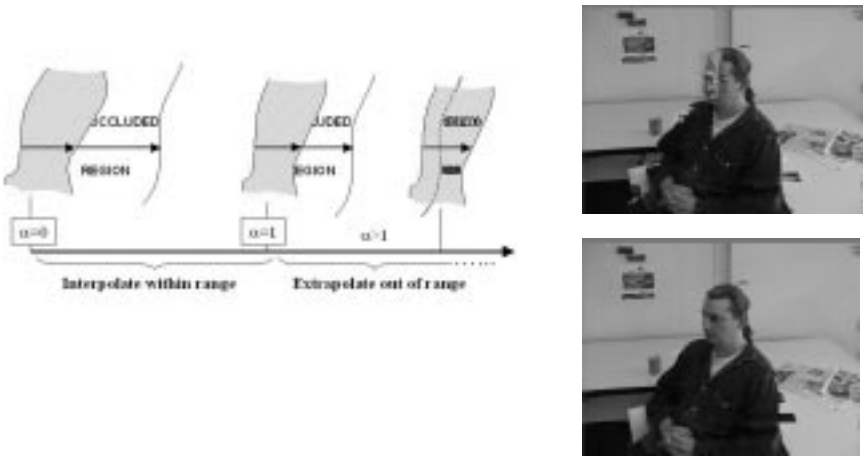


Figure 4. Left: Diagram showing occlusion and interpolation beyond range. Right: Example of undesirable edge fragmentation while extrapolating beyond range (top) and correction by depth of field ordering (bottom).

raster whose projection into the three views satisfies order with respect to the selected edge and for which the line joining the projection of the two points in each view is free from other edge intersections.

An example is shown in Figure 4 (right), where two versions of a rendered image from a viewpoint beyond range given by (α=1.5, β=0) are shown with no correction (top) and with correction (bottom). Without correction, the order violations of the interpolated edges results in unsightly fragmentation, particularly noticeable between the head and features from the background wall. By applying the heuristic just described, the corrected image on the right is free from distortion.

6 Experimental Results

The method presented in this paper has been applied to a number of image triplets of real scenes. The results obtained very encouraging given the complexity of the scenes and that the method is fully automatic.

Figure 4 shows an example of synthesized viewpoints with the respective values of α and β overlaid. Note that this example shows images interpolated beyond range. For reason of space, it is impossible to include other examples here but more can be found on http://hplbwww.hpl.hp.com/people/mp/research/edgeibr/index.htm where higher resolution images are also available

The method is, due to its nature, very dependent on the quality of edge matching and in some examples we have noticed a number of artifacts, due to errors in the edge matching process. Hence any improvements in edge matching are very desirable.

Time-wise, the current non-optimised implementation of the matching part runs in

about 30 seconds on an HP-9000™ workstation. The rendering can be achieved under Windows™ on a 200MHz Pentium™ PC at a rate of over ten 640x480 frames a second.

Figure 5. Snapshots from the renderer. All images are synthesized viewpoints and the three vertices, in particular, are extrapolated beyond the three primary images.

7 Conclusion

This paper has presented a novel method that relies on computer vision techniques to perform automatically image based rendering.

There are two novel aspects of the work. First and above all, for the first time edges matches have been shown to be an efficient representation for generic view synthesis. They give a major advantage over previous methods as they are easier and faster to match and result in an efficient rendering scheme. As a consequence of using edge transfer for view synthesis we have applied the method to three images, a definite novelty for methods relying on sparse correspondences. We have also been able to extend the method to deal with occlusions.

The results so far are very encouraging but some artefacts are nonetheless present due to edge mismatches. We are currently investigating better edge matching strategies.

The method has also been successfully applied to a number of short video sequences of animated objects. The principle of the proposed method is not short of applications in immersive videoconferencing, 3D sprites for the WEB and compression.

APPENDIX. 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 where $f_\alpha(\alpha)$

$$\mathbf{A}_s \approx \mathbf{A}_0\left((1-f_\beta(\beta))((1-f_\alpha(\alpha))\mathbf{v_0} + f_\alpha(\alpha)\mathbf{v_1}) + f_\beta(\beta)\mathbf{v_2}\right)$$

and $f_\beta(\beta)$ 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 $\mathbf{v}_2$.

References

[1] Avidan,S. & Shashua, A. "Novel View Synthesis in Tensor Space", *Proceeding of Computer Vision and Pattern Recognition Conference 1997*, pp 1034-1040, 1997.

[2] Beymer, D. Shashua, A. & Poggio, T. "Example Based Image Analysis and Synthesis", MIT AI-Lab. Tech. Memo No. 1431, 1993.

[3] N.L.Chang & A. Zakhor, "View Generation for Three-dimensional Scenes from Video Sequences", *IEEE Trans. on Image Processing*, Vol 6, No 4, April 1997.

[4] Chen, S.E., "QuickTime® VR- An Image-Based Approach to Virtual Environment Navigation", *Proc. SIGGRAPH 95*. In *Computer Graphics*, pp29-38, 1995.

[5] Chen, S.E. and Williams, L. "View interpolation for image synthesis", , *Proc. SIGGRAPH 93*. In *Computer Graphics*, pp279-288, 1993.

[6] Greene, N., "Environment Mapping and Other Applications of Word Projections", *IEEE Computer Graphics and Applications*, Vol 6, No 11, pp 21-29, 1986.

[7] Gortler, S.J, Grzeszczuk, Szeliski, R. & Cohen, M.F., "The Lumigraph", *SIGGRAPH 96, In Computer Graphics*, pp 31-42, 1996.

[8] 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.

[9] Laveau, S & Faugeras, O., "3D Scene Representation as a Collection of Images and Fundamental Matrices", INRIA Tech Report 2205, February 1994.

[10] McMillan, L. & Bishop, G., "Plenoptic Modelling". *Proc. SIGGRAPH 95, In Computer Graphics*, pp 39-46, 1995

[11] Ohta, Y. & Kanade, T., "Stereo by Intra- and Inter-Scanline Search", *IEEE Trans. PAMI*, Vol. 7, No. 2, pp139-154, 1985..

[12] 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

[13] Shapiro, L.S., *Affine Analysis of Image Sequences*, Cambridge University Press, 1995.

[14] Szeliski, R., "Video Mosaics for Virtual Environments", *IEEE Computer Graphics and Applications*, 22-30, March 1996.

[15] Ullman, S. & Basri, R., "Recognition by Linear Combinations of Models", *IEEE Trans. PAMI*, Vol 13, No 10, pp 992-1006, 1991.

[16] Wolberg, G., *Digital Image Warping*, IEEE Computer Society Press, 1990.

[17] Yachida, M., "3D Data Acquisition by Multiple Views", *Third International Symposium of Robotics Research*, Faugeras, O.D. & Girault, G. Eds, MIT Press, 1986.

[18] 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.