Reconstruction of Visual Appearance

Peter R.J. North
School of Cognitive and Computing Sciences
University of Sussex
Janet: petern@cogs.susx.ac.uk

This paper presents a method for reconstructing the visual appearance of a scene from any viewpoint, given a collection of stereo pairs. A representation is suggested which is flexible and efficient, and algorithms to recover three dimensional structure and grey-level detail by combining a number of two-dimensional views are discussed. Some results using real and simulated data are given.

This paper presents a method for reconstructing the visual appearance of a scene from any viewpoint given a collection of stereo pairs. For example, figure 1 shows two stereo pairs of the same building. The objective is to generate a new view from this data as in figure 2 (see below). The method has a number of properties:

- Occlusion is dealt with explicitly and all surface fragments visible from at least one stereo pair are correctly rendered in the output.
- Surface shape and texture are obtained from the best available data.
- The stereo pairs are pre-compiled into a concise representation from which reconstruction is an $O(n)$ process (n = number of stereo pairs).

Reconstruction of visual appearance has been used extensively for visualisation of terrain models [1]. Some authors have also used reconstruction to show the result of stereo algorithms [2]. The Mosaic system [3] used stereo and geometric reasoning to generate wire-frame models of urban scenes. Although domain specific to aerial photographs of buildings, some results using grey-level rendering were achieved. Feature tracking [4, 5, 6, 7] has been used to recover three-dimensional structure on a frame-by-frame basis. The De-Launay triangulation [8] is used here [4] and elsewhere [2] in reconstruction and representation, and has the advantage of preserving discontinuities. Although some reconstruction of visual appearance was performed with this representation we believe detailed grey-level mapping is made difficult by the fractal nature of the projected surface. Octree models allow efficient updating by volume intersection techniques, and have been used for model generation under controlled conditions [9, 10]. However the representation is inflexible with respect to scale, and expensive for storing much grey-level detail. Potmesil [11] describes a method of merging surface segments by minimising shape differences. However the domain is restricted to model generation of continuous objects, with accurate depth information using a controlled light source.

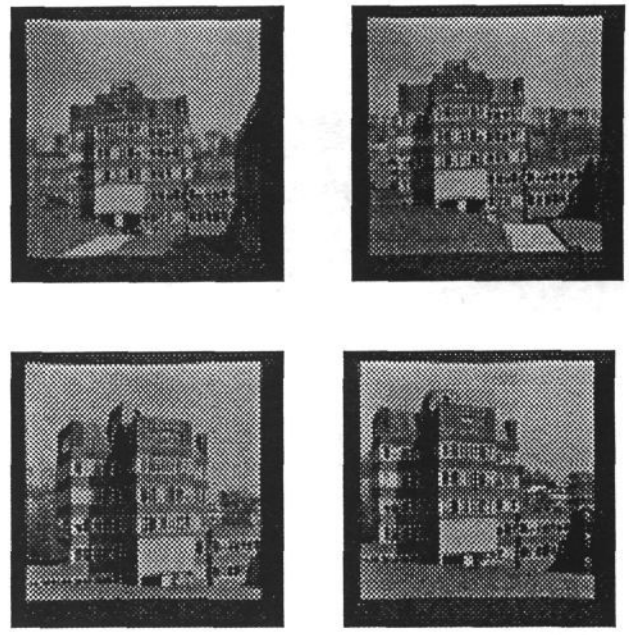

Figure 1: **Stereo pairs.**

SYSTEM OVERVIEW

The system works in two parts. Steps 1, 2, 3 and 4 are performed off-line from the available stereo pairs. Step 5 is performed for each reconstruction required.

1. Production of a depth map from each stereo pair.
   The relative camera orientation is determined and corresponding points are found for each pair. These are projected by triangulation to give a grid of three-dimensional locations.
2. Computation of relative positions of stereo pairs.
   Overlapping regions of the surfaces generated from the stereo pairs allow the relative positions of their viewpoints to be computed.
3. Detection of depth map discontinuities.
   By comparing depth maps, discontinuities in the interpolated depth maps are located and marked to avoid unwanted interpolation during reconstruction. The result is a set of viewpoint-dependent depth maps with discontinuities, somewhat similar to Marr's concept [12] of a $2\frac{1}{2}d$ *sketch*.

BMVC 1990 doi:10.5244/C.4.37

Figure 2: **Rendered view.**

4. Detection of highest resolution surface patches.
   The depth maps are compared so that information on any surface fragment is stored only once, in the depth and texture maps containing the most detailed description of it.

5. Reconstruction of new view.
   This is achieved by rendering each of the surfaces from the new viewpoint, with proper treatment of hidden surfaces and discontinuities.

DEPTH MAP GENERATION

In order to use the epipolar constraint, and in order to locate points in three dimensions once correspondences have been found, it is necessary to determine the *relative camera orientation.* This means finding the rotation matrix R and translation vector T which map the coordinate frame of the first camera position to that of the second. Algorithms exist [5, 13, 14] to recover this from a small set of image correspondences, subject to a scale factor in T. In the current work, these correspondences are entered manually, and the relative camera orientation is determined using an iterative least-squares method [15] [1] on a minimum of eight points.

Once the relative camera orientation is known, the *epipolar constraint* can be used to assist the correspondence algorithm. A point in one image will project as a line in three-space, and hence as a line in the other image. The subsequent search for the corresponding point is therefore confined to this line, reducing search time and decreasing the likelihood of false matches.

The *correspondence* algorithm used is based on Nishihara's stereo matching system [17]. The images are first convolved with a difference of gaussians filter, and the sign of the result at each image location is stored. Correlation matching then takes place along epipolar lines in the resulting image pair. The correlation surface near its peak is approximated as a cone, and a coarse-to-fine strategy is employed, enabling the location of maximum correlation to be found by just a few measurements.

The output from the correspondence algorithm is a grid of image disparities. Having determined R and T we may now calculate *three-dimensional coordinates* of these points, using equations [13]:

$$Z = \frac{(R_1 - x'R_3) \cdot Tf}{(R_1 - y'R_3) \cdot x}$$

where R_i is the ith row of R, and

$$X = Zx/f, \qquad Y = Zy/f$$

for camera focal ratio f, and image coordinates (x, y) and (x', y'). In fact only the Z coordinate need be stored in order to recover three dimensional structure at a later stage. Regularisation takes place using gaussian interpolation (approximately equivalent to spline interpolation [18]) to produce a regular depth map.

VIEWPOINT REGISTRATION

The relative positions of the depth maps must now be determined. Each pair of depth maps are related by unknown rotation R, translation T and scale λ parameters,

These are determined from corresponding three-dimensional points u_i and v_i by minimising the error E given by

$$E = \sum_i \mid \lambda R \cdot (u_i - T) - v_i \mid^2$$

using the downhill simplex method [16].

Exact relative positions are assumed known for the artificial stereo pairs (figure 7).

DISCONTINUITY LOCATION

For successful reconstruction it is important to identify depth discontinuities. On surface rendering, it is possible for the interpolation resulting from a depth discontinuity to give rise to unwanted occlusion. Figure 3 shows a cross-section of the surface resulting from reconstruction from two depth maps, A and B. Some surface reconstructed from A is occluded by an interpolated region from B, and *vice versa.* The objective is to enter a zero into all depth maps at places where there is such a depth discontinuity. To achieve this the surface represented by each depth-map is rendered into all other depth maps.

Surface rendering is achieved using bi-linear splines to interpolate depth and grey-level data, using the depth data and grey-level data as control points.

The spline surface is divided into a regular tesselation of triangular facets (not necessarily at the same resolution as the underlying control points). Each facet is then rendered directly into the target image selecting grey-levels

[1] An attempt was made using an implementation of a direct solution [13] but was found to be unstable with respect to noise.

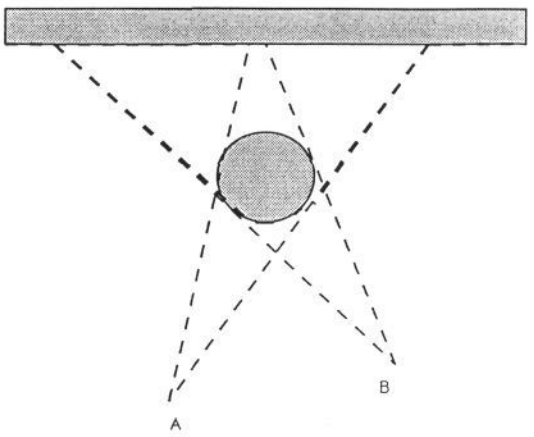

Figure 3: **Occluding interpolated surface.**

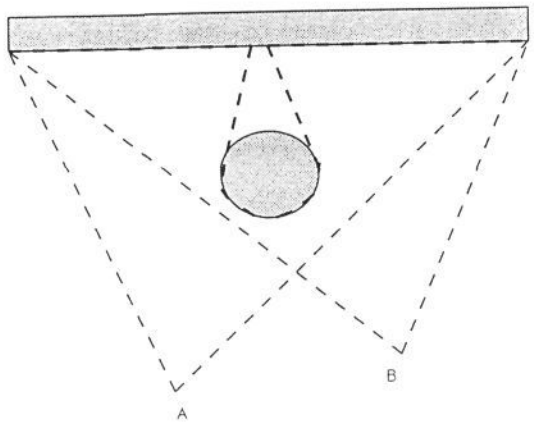

Figure 4: **Result of algorithm 1.**

from the grey-level surface via interpolation of the underlying parameters. Depth-buffering is used to ensure correct treatment of hidden surfaces.

To locate occluding discontinuities we note that the interpolated surface lies between the viewpoint and a surface visible from some other depth map.

Hence, for each depth map, render it into depth maps derived from other stereo pairs. Where a rendered value appears closer than the value in any other depth map, a zero is entered into the first depth map to indicate occlusion.

Algorithm 1:

For each depth-map D_i

For each other depth-map D_j

- Render each point (x, y) in D_i from the viewpoint of D_j.
- Compare the result (x', y', z') with depth map D_j.
- If any location $D_j(x', y')$ has a greater value than z', then mark $D_i(x, y)$ as a discontinuity.

This algorithm is analogous to volume-intersection techniques used to build octree models. Results of operation on a synthetic set of depth-maps are shown (figure 7). The left images show three perspective views of a box 'floating' in the center of a square room, similar to that illustrated in cross-section (figures 3,4 and 6). Distance from the center of projection is illustrated as intensity. The right-hand images show the discontinuities found marked in black.

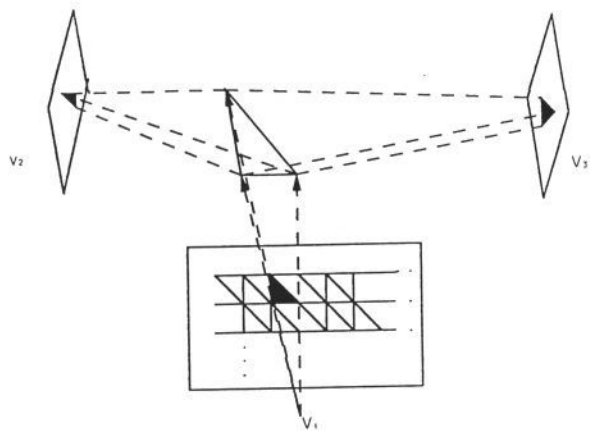

Figure 5: **Overlapping surfaces.**

OVERLAPPING SURFACES

When a surface patch is represented in more than one depth map it is necessary to select the data to be used in rendering. Rather than seek to combine information from more than one source we use the highest resolution data unchanged.

This is determined by noting that a triangle projected from one viewpoint will increase in area if back-projected onto another view-plane with higher resolution information (figure 5).

Algorithm 2

For each depth map D_i

For each other viewplane V_j

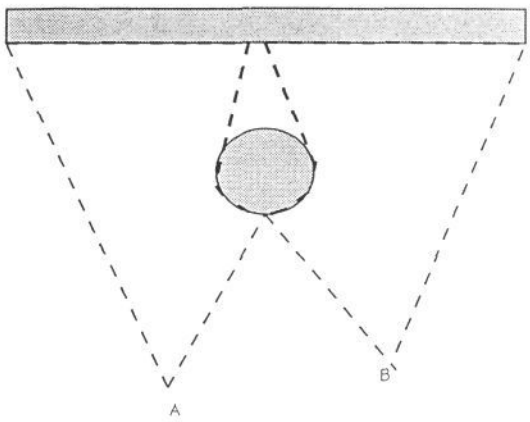

Figure 6: **Final representation.**

- Project each triangle of D_i onto the viewplane V_j
- Calculate the new area A' of this triangle
- If $A' > A$, the original area, for *any* viewplane V_j, then mark the triangle as a hole

The operation results in the final surface description illustrated in figure 6.

RECONSTRUCTION

Each surface is rendered using the triangular tesselation as above. A hole or discontinuity in a surface is marked as a zero in its depth map. A triangle containing such a hole is not rendered, hence giving the appearance of transparency in these regions of the surface.

Some results of the reconstruction process are shown (figures 7, 8 and 9) from the stereo pairs given before. Note that these reconstructions have been produced directly from their respective stereo pairs without registration and discontinuity location. Figures 10 and 11 show the results of registration, with surfaces from each stereo pair approximately overlapping.

DISCUSSION

A method has been presented for a system to automatically construct a composite model of an arbitrary scene, complete with grey-level information, from a set of two-dimensional images. A representation has been proposed which models scene discontinuities and preserves the optimal grey-level detail for all surfaces present. An algorithm to locate such discontinuities by combining information from new viewpoints has been described. Finally some results on real and synthetic images are given.

Further work is intended in the following areas.

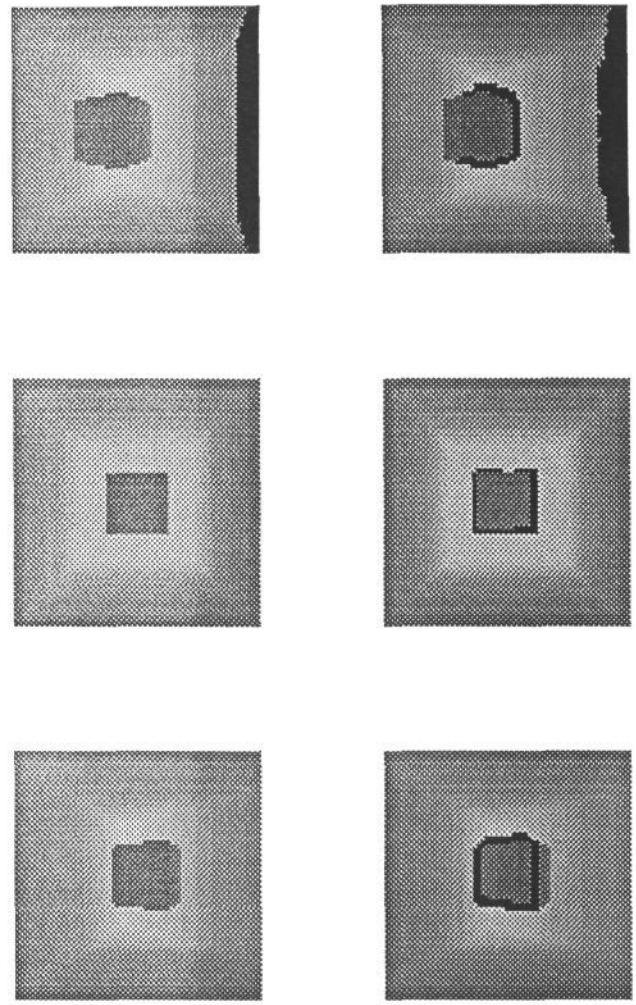

Figure 7: **Discontinuity location.**

- Although it is believed best to take grey level detail from only one surface segment in the case of overlap, an improvement would be to utilise the new geometrical information to reduce error. Some researchers, [4, 6, 7] have shown success using Kalman filtering to model and reduce error over time, usually in tracking image features. It would be interesting to explore this technique in modeling uncertainty in surface location.
- The algorithm presented illustrates one way to locate depth discontinuities. However it may be possible to locate these directly during surface interpolation, for example using the weak plate model [19].
- The correlation-based stereo algorithm used performs poorly at surface boundaries, and areas of high gradient where there is significant shearing of the image between viewpoints. Further work is needed to prevent this.

ACKNOWLEDGEMENTS

Firstly I would like to thank David Hogg for his help, and for comments on a draft of this paper. Thanks also to Harry Barrow, Alistair Bray, Christopher Longuet-Higgins and Kelvin Yuen for helpful discussions.

Figure 8: **Rendered view from first stereo pair.**

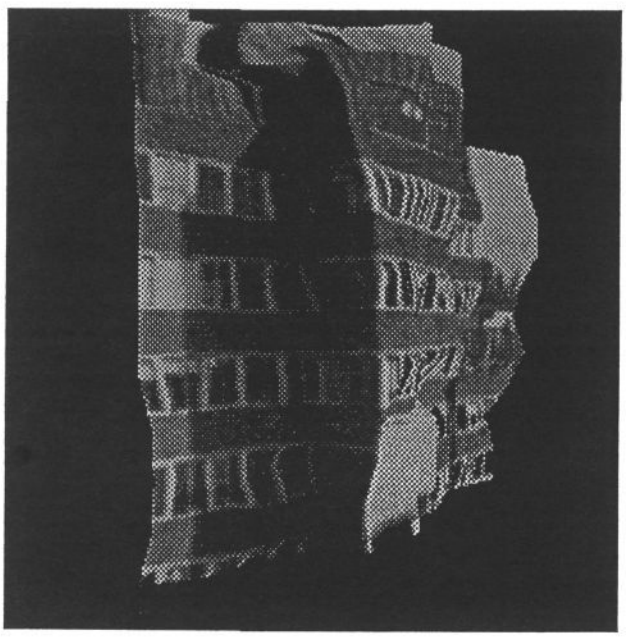

Figure 9: **Rendered view from second stereo pair.**

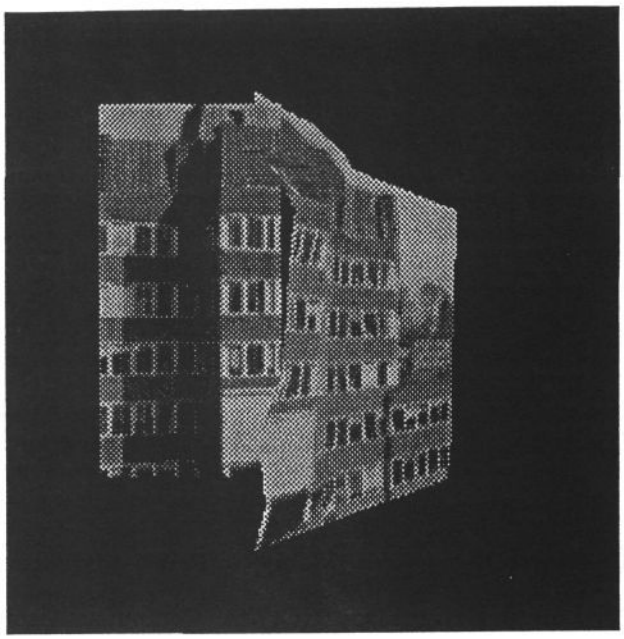

Figure 10: **Rendered view showing registration.**

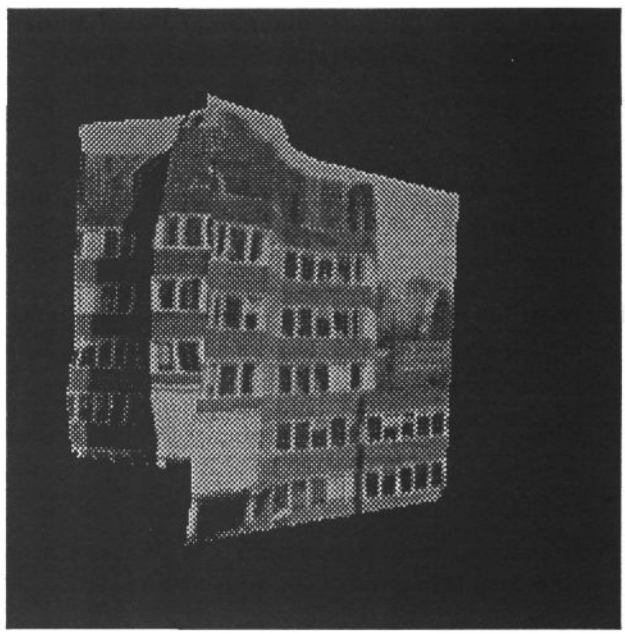

Figure 11: **Rendered view showing registration.**

References

[1] **Day, T. and Muller, J-P.** "Digital Elevation Model Production by Stereo-matching SPOT Image-pairs: A Comparison of Algorithms" *AVC*, Manchester, pp117-122, August 1988.

[2] **Le Bras-Mehlem, E., Schmitt, M., Faugeras, O.D. and Boissonat, J.D.** "How the Delaunay Triangulation Can Be Used for Representing Stereo Data". *Second Int. Conf. on Computer Vision*, December 5-8, 1988, Florida, pp. 54-63.

[3] **Herman, M. and Kanade, T.** "Incremental Reconstruction of 3D Scenes from Multiple, Complex Images". *Artificial Intelligence* Vol. 30, No. 3, pp.289-341, December 1986.

[4] **Charnley, D. and Blisset R.** "Surface Reconstruction from Outdoor Image Sequences". *AVC*, Manchester, pp. 153-158 ,August 1988.

[5] **Harris, C.** "Determination of Ego-Motion from Matched Points". *AVC*, Cambridge, September 1987.

[6] **Matthies, L., Szeliski, R. and Kanade, T.** "Kalman Filter-based Algorithms for Estimating Depth from Image sequences". *DARPA Image Understanding Workshop* 1988, pp. 199-212, April 1988.

[7] **Stephens, M. and Harris, C.** "3D Wire-Frame Integration from Image Sequences". *AVC*, Manchester, pp. 159-165, August 1988.

[8] **Boissonnat, J.D.** "Representing 2D and 3D Shapes with the Delaunay Triangulation". *Proc.*

of the 7th Int. Conf. on Pattern Recognition, Montreal, Canada,pp. 745-748, 1984.

[9] **Connolly, C.I. and Stenstrom, J.R.** "Generation of Face-Edge-Vertex Models Directly from Images". *DARPA Image Understanding Workshop* 1988, pp. 1041-1046.

[10] **Hong, T. and Shneier M.** "Incrementally Constructing a Spatial Representation Using a Moving Camera". *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 19-23, 1985, San Francisco, pp.591-596.

[11] **Potmesil, M.** "Generating Models of Solid Objects by Matching 3D Surface Segments". *CAI* 1983, pp. 1089-1093.

[12] **Marr,D.** *Vision*, Freeman, San Francisco, 1981.

[13] **Longuet-Higgins, H.C.** "A Computer Algorithm for Reconstructing a Scene from Two Projections". *Nature*, 293, pp. 133-135, 1981.

[14] **Ullman, S.** *The Interpretation of Visual Motion*, MIT Press, Cambridge, Massachusetts, 1979.

[15] **Longuet-Higgins, H.C.** Personal communication, 1989.

[16] **Press, W., Flannery, B., Teukolsky, S. and Vetterling, W.***Numerical Recipes in C* Cambridge University Press, 1988.

[17] **Nishihara, H.K.** "Practical Real-time Imaging Stereo Matcher". *Optical engineering*, 23(5), pp536-545, September-October 1984.

[18] **Li-Dong Cai** "Spline Smoothing: A Special Case of Diffusion Smoothing". *AVC* 1989, Reading, pp. 273-276.

[19] **Blake, A. and Zisserman, A.** *Visual Reconstruction.* MIT press, Cambridge USA, 1987.