

Line Based Trinocular Stereo

D. Yang and J. Illingworth

Dept. of Electronic and Electrical Engineering,

University of Surrey,

Guildford, Surrey GU2 5XH, United Kingdom

Abstract

An approach to solving the stereo correspondence problem in trinocular stereo vision is described. It is based on geometric matching constraints relating the orientation of lines extracted in three images taken from different viewpoints. These novel constraints are termed *unary orientation* and *binary orientation* constraints. Matching is achieved within an optimisation framework in which the constraints are encoded into a cost function that is optimised using the *simulated annealing* method. Results are demonstrated and the characteristics of the approach are explored on both synthetic and real¹ trinocular images.

1 Introduction

A fundamental problem in computer vision is the inference of 3D structure from 2D images. An important machine vision technique for this is stereo vision. In stereo vision several images of a scene are taken from different viewpoints and an attempt is made to find the correct correspondence of image features that are projections of the same physical entity. If the geometric relationship between viewpoints is well known then accurate 3D structure can be recovered using the measured image disparity (or shift) between matched features. The problem addressed in this paper is that of finding the correct feature correspondences in a set of three stereo images.

Previous work in stereo vision has involved matching of point or line features. Line matching schemes in binocular and trinocular vision [1, 2, 3, 6, 7, 8] have been primarily based both on similarity of feature attributes and on using point-based representations of lines (midpoints or endpoints) that allow adoption of epipolar constraints developed for point based stereo matching. Unfortunately, matching based on similarity of features is limited to cases where the angle between views is small, while point-based matching is not robust as occlusion effects or line fragmentation artifacts mean that midpoints or endpoints may not correspond to the same 3D point in a scene.

In this paper we attempt to exploit fully the geometric constraint information inherent in lines among three views. We suggest more direct geometric constraints based on the relationship between the orientation of lines in three distinct images. Two main results are exploited: the first uses the orientation

¹The real trinocular data was kindly supplied by Dr N. Ayache of INRIA, Rocquencourt, France and is the data used in his recent book "Artificial Vision for Mobile Robots: stereo vision and multisensory perception"

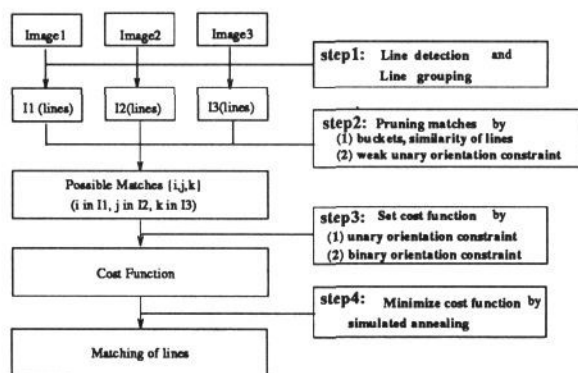


Figure 1: Overview of the Algorithm

of lines in three images and is called the *unary orientation* constraint while the second constraint relates pairs of lines matched in three images and is called the *binary orientation* constraint. These constraints are viewpoint independent and it is the distinctive feature of our method. These constraints are encoded into an optimisation matching algorithm.

An overview of the proposed algorithm is shown in Figure 1. Lines segments are extracted in three images taken from different viewpoints. Segments which are collinear are grouped into a single line as our algorithm primarily uses only orientation information. Each line segment is represented by an N-vector [4], which is the unit normal to the plane that contains the image line and the optical centre of the viewing camera. The second stage of the algorithm is a simple pruning step that discards inadmissible matches using heuristics such as bucketing of lines and line similarity information which are commonly used by other authors [1, 2, 3, 6, 7, 8]. In addition, the proposed weak unary orientation constraint is also used in this step. Note that they are applied with loose thresholds so that they throw out wrong matches but retain the subset that includes the correct matches. The objective of the second stage is to quickly reduce the size of the matching problem. In the third step of our method the proposed unary and binary orientation constraints are encoded into a cost matching function. The configuration of matches represented by the minimum of the cost function correspond to the best correspondence of the image lines. Any suitable optimisation algorithm could be used for minimisation but in this work we have used simulated annealing in order to find a global minimum. The computational cost of the process kept manageably low as the number of matches has been reduced by the preceding grouping and pruning stages.

The rest of the paper is organised as follows: section 2 discusses in detail the geometric constraints for the orientation of lines viewed from three different views and derives the unary and binary constraints. Section 3 shows how the constraints may be encoded into a matching cost function and briefly discusses the simulated annealing method that is used to find the minimum cost solution. Section 4 shows experimental results of tests of the algorithm on both real and synthetic data while Section 5 summarises the contribution of the work and offers suggestions for its future development.

2 Line Orientation Constraints

In this section, the representation of 2D and 3D lines is considered and the relation between 2D lines and 3D lines is established. Geometric constraints for orientation of lines among three different views are explored. Two kinds of orientation constraints for matching lines in three images are introduced: *unary* and *binary* orientation constraints.

The camera model used is the pinhole model and perspective projection is adopted as the model of the image formation process. The relation between a 3D point and its 2D projection is expressed by the following:

$$\begin{pmatrix} \omega u \\ \omega v \\ \omega \end{pmatrix} = T \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

where (u, v) are image coordinates, ω is a scalar, (x, y, z) are 3D point coordinates, T is a 3×4 matrix called the perspective transformation matrix and is defined up to a scale factor.

Points and lines in the image plane are represented (uniquely up to a sign) by unit vectors of homogeneous coordinate called *N-vectors* [4]. The unit vector starting from the optical center and pointing toward a point P in the image plane is called the *N-vector of point P*. The unit vector normal to the plane passing through the optical center C and intersecting an image plane along a line l is called the *N-vector of line l*. Figure 2(a) shows an example of representation of a point and a line by N-vectors.

For 3D vectors \mathbf{a} , \mathbf{b} and \mathbf{c} the following notations are used throughout the paper: $\mathbf{a} \cdot \mathbf{b}$ denotes the inner or dot product, $\mathbf{a} \times \mathbf{b}$ denotes the cross product, $\langle \mathbf{a}, \mathbf{b}, \mathbf{c} \rangle = (\mathbf{a} \times \mathbf{b}) \cdot \mathbf{c} = (\mathbf{b} \times \mathbf{c}) \cdot \mathbf{a} = (\mathbf{c} \times \mathbf{a}) \cdot \mathbf{b}$ denotes the scalar triple product, $[\mathbf{a}] = \mathbf{a}/\|\mathbf{a}\|$ denotes the normalization of vector \mathbf{a} and $\|\mathbf{a}\|$ denotes the norm of vector \mathbf{a} .

The N-vector, \mathbf{m} , of a point $P(u, v)$ can be obtained from the perspective projection matrix T as $\mathbf{m} = [(\mathbf{t}_1 - u\mathbf{t}_3) \times (\mathbf{t}_2 - v\mathbf{t}_3)]$ where the vector $\mathbf{t}_i = (t_{i1}, t_{i2}, t_{i3})^t$ and t_{ij} is an element of T . The N-vector, \mathbf{n} , of a line l is then simply given by the cross product of the N-vectors of any two points which lie on the image line i.e. $\mathbf{n} = \mathbf{m}_1 \times \mathbf{m}_2$.

A 2D line and the optical centre define a plane in 3D space and the N-vector describes the normal to that plane. If a line is taken from each of two distinct views then the two planes generated may intersect in 3D to define a 3D line. The equation of the 3D line is easily calculated from the N-vectors, $\mathbf{n}_1, \mathbf{n}_2$ of the two lines. The 3D scene line can be parameterised by a unit direction vector \mathbf{v} and the foot of the normal point, Q , on the 3D line. \mathbf{v} and Q can be determined using the relations (see Figure 2(b)):

$$\mathbf{v} = \mathbf{n}_1 \times \mathbf{n}_2, \quad O\vec{Q} \cdot \mathbf{v} = 0, \quad Q\vec{C}_1 \cdot \mathbf{n}_1 = 0, \quad Q\vec{C}_2 \cdot \mathbf{n}_2 = 0$$

However, taking a line from each of two views is not a sufficient condition to guarantee the existence of a physical line in 3D. Any matched pair of lines will generate a hypothesis for a 3D scene line but it can only be verified as a

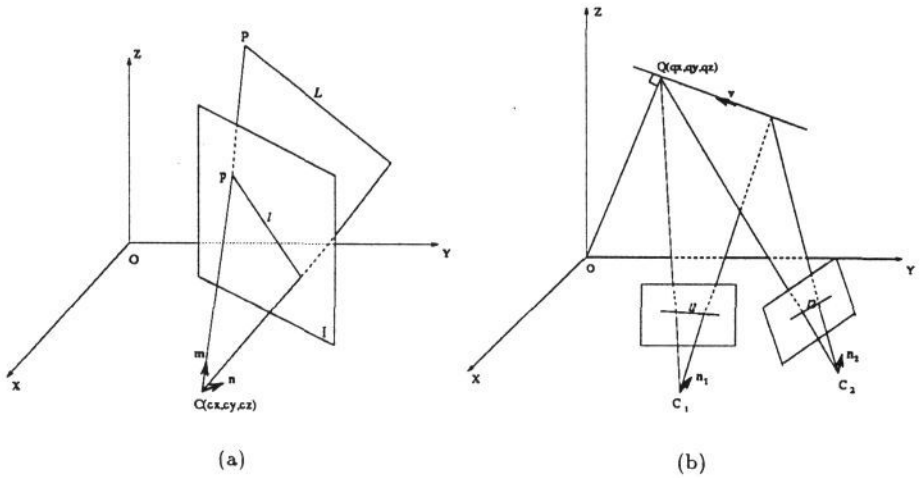


Figure 2: (a) N-vector \mathbf{m} representing a point p and \mathbf{n} representing line l on the image plane I , where C is the optical center, P and L are the corresponding 3D point and line, respectively (b) Reconstruct 3D line by two views

physically realisable line by finding a match with a geometrically consistent line in a third view. The geometric constraints on lines in three views will now be explored and used to define criteria for identifying correct line correspondences.

Consider images taken from three different viewpoints in which lines can be represented by N-vectors as shown in Fig. 3. If the N-vectors of the projection of a given 3D line in each of the 3 views are denoted as \mathbf{n}_1 , \mathbf{n}_2 and \mathbf{n}_3 then two unary orientation constraints (so called as they relate to the match of a single 3D line) can be defined as follows:

Weak Unary Orientation Constraint:

$$f_{wu}(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3) = \langle \mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3 \rangle = 0 \quad (1)$$

This constraint is called a weak constraint as it is a necessary but not sufficient condition for a correct match. It is used to discard inadmissible matches in the pruning step in our scheme. A single match $\mathbf{m} = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$ is correct iff a pair of 2D lines $\{\mathbf{n}_1, \mathbf{n}_2\}$ reconstruct the same 3D line as a pair of 2D lines $\{\mathbf{n}_2, \mathbf{n}_3\}$. Thus a stronger unary constraint is:

Unary Orientation Constraint:

$$f_u(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3) = (Q_{12}x - C_3x, Q_{12}y - C_3y, Q_{12}z - C_3z) \cdot \mathbf{n}_3 = 0 \quad (2)$$

where Q_{12} denote the foot of the normal on the 3D line reconstructed by $\{\mathbf{n}_1, \mathbf{n}_2\}$ and C_3 is the viewpoint of the third camera (see Figure 3).

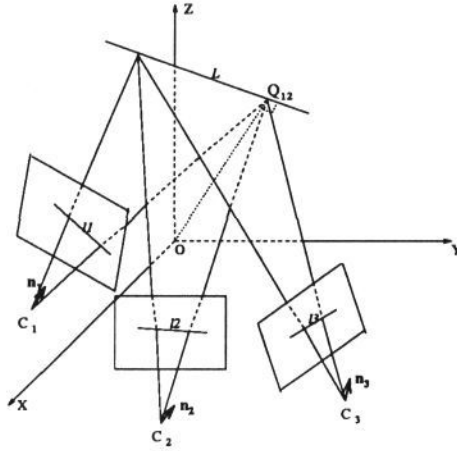


Figure 3: N-vectors representing lines in trinocular vision

Further constraints can be found by considering the geometric relation for a pair of matches $m_1 = \{n_{i1}, n_{j1}, n_{k1}\}$ and $m_2 = \{n_{i2}, n_{j2}, n_{k2}\}$. A 3D line can be constructed by two 2D lines in different viewpoint images. For three views, there are two independent combinational two 2D lines, say, (n_{i1}, n_{j1}) and (n_{j1}, n_{k1}) . Since m_1 is supposed to be a correct match the constructed 3D lines $v_1 = [n_{i1} \times n_{j1}]$ and $v_2 = [n_{j1} \times n_{k1}]$ are supposed to be the same line. Similarly, the 3D line $u_1 = [n_{i2} \times n_{j2}]$ and $u_2 = [n_{j2} \times n_{k2}]$ are supposed to be the same line. Thus, the angle between line v_1 and line u_1 should be equal to the angle between line v_2 and line u_2 . Thus we have the following binary constraint (so called as it involves pairs of matches):

Binary Orientation Constraint:

$$\begin{aligned}
 f_b((n_{i1}, n_{j1}, n_{k1}), (n_{i2}, n_{j2}, n_{k2})) \\
 &= ([n_{i1} \times n_{j1}] \cdot [n_{i2} \times n_{j2}]) - ([n_{j1} \times n_{k1}] \cdot [n_{j2} \times n_{k2}]) \\
 &= 0
 \end{aligned} \tag{3}$$

This constraint can be used to enforce mutually support between correct matches.

It is worth noting that the proposed orientation constraints are viewpoint independent. This is in contrast to other methods that use orientation information primarily as a similarity measure for matching. Similarity approaches only work where the angle between views are small and are therefore viewpoint dependent. The proposed orientation constraints make extensive use of the third view and are valid for all viewing positions.

It is also worth noting that the proposed constraints are local constraints. in the next section we discuss how they can be incorporated into a matching framework where the correct matches are determined by the global minimum of a cost function. The minimum is found by making local changes to search through the solution space.

3 Matching as Optimization

In this section, we concentrate on the representation of the solution space for the three view matching problem and then formulate a cost function which aggregates local costs to quantify the global goodness of the solution. The three view stereo matching problem then corresponds to locating the minimum cost solution and this is achieved via local changes using simulated annealing.

A $N_1 \times N_2 \times N_3$ binary matrix \mathbf{P} , where N_1, N_2 and N_3 are respectively the numbers of lines in each of the three images, is used to represent a solution of the stereo matching problem. An element of \mathbf{P} is defined by:

$$p(i, j, k) = \begin{cases} 1 & \text{if } (i, j, k) \text{ is a match} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Since a line in an image can match at most one line in the other images or no lines due to occlusion, \mathbf{P} is constrained to have the following property:

$$\sum_{i=1}^{N_1} p(i, j, k) = 1 \text{ or } 0, \quad \sum_{j=1}^{N_2} p(i, j, k) = 1 \text{ or } 0 \text{ and } \sum_{k=1}^{N_3} p(i, j, k) = 1 \text{ or } 0 \quad (5)$$

In principle, the solution space of the problem, \mathbf{S} , is the set of all possible values of the matrix \mathbf{P} subject to the constraint defined by the above equation. However this is extremely large and therefore it is necessary to reduce the size of \mathbf{S} by pruning the set of lines and matches using simple heuristics based on factors such as grouping of collinear segments in a single image and applying loose similarity constraint and weak orientation constraint to exclude infeasible matches between images.

The cost of a particular feasible solution can be expressed a linear sum over all admissible matches in solution space of two terms:

$$E(\mathbf{P}) = \lambda_1 \sum_{p \in P} p(i, j, k) E_1(i, j, k) + \lambda_2 \sum_{p_1 \in P} \sum_{p_2 \in P (p_1 \neq p_2)} p_1(i_1, j_1, k_1) p_2(i_2, j_2, k_2) E_2((i_1, j_1, k_1), (i_2, j_2, k_2)) \quad (6)$$

The first term relates to the degree of satisfaction of the unary orientation condition while the second term relates to the binary orientation constraint. λ_1 and λ_2 are weight factors controlling the relative influence of the two constraints. In our work so far we have used $\lambda_1 = \lambda_2 = 0.5$ throughout although this choice warrants further detailed investigation. The particular form adopted for energy terms, E_1 and E_2 , is

$$E_1 = \exp\left(\frac{|f_u|}{\sigma_1}\right), \quad E_2 = \exp\left(\frac{|f_b|}{\sigma_2}\right) \quad (7)$$

where σ_1 and σ_2 is the control parameters set to values related to the expected variance of the values of the orientation constraints. In all our work the values of σ_1 and σ_2 were fixed.

In order to find the minimum cost solution of the matching function we have applied the method of simulated annealing [5]. This is a stochastic search method

in which state changes that lead to cost decreases are always accepted but local changes that result in a cost increase may be accepted with a small random probability. This strategy allows the search to jump out of local minimum and asymptotically guarantees convergence to the global minimum. The probability with which cost increases are accepted is varied as the process continues, being initially high but then decreasing later. The parameter that controls this probability is known as the “temperature” following the analogy of annealing in physical systems.

4 Experimental Results and Discussion

In this section results are presented of the algorithms performance on both synthetic images and real trinocular stereo data. All algorithms were implemented in C and run on a Sun Sparcstation.

Synthetic images were generated using the Rayshade software package. This permits scenes composed of simple geometric primitives to be constructed and allows explicit control and knowledge of factors such as viewing parameters and noise. Parts (a), (b) and (c) of Figure 4 shows a typical triplet of images of the corner of a laboratory. These images were processed by an implementation of the Canny edge detector and lines were extracted using a Hough Transform. It should be noted that matching methods based on line attribute similarity would fail to match several of the lines in these images as the viewpoint changes are relatively large. The results of the algorithm for this test image are shown in parts (d) through (f) of Figure 4, where correct matches are shown via line numbers placed adjacent to the lines i.e. line number 1 in (d) is matched to line number 1 in (e) and line number 1 in (f). In this example all 15 lines are successfully matched. On a larger selection of synthetic images the average correct matching percentage was about 90%. Incorrect matches can generally be attributed to not having obtained a global minimum in optimization. The computation time for these images was less than 1.5 minute of CPU.

To test the robustness of our method synthetic image data was corrupted by adding uniform random noise to each of the three components of the original line N-vectors. The amount of noise added to each component was a constant fraction of the size of the component. The method was found to work reliably up to values of noise where the maximum deviation in orientation was 2.5 degrees. The average correct matching percentage was about 85%. We found that presence of noise in these cases results in more local minimums in cost function, but global minimum of cost function still corresponds to the best matching. It means that the proposed unary and binary orientation constraints have a fairly stringent tolerance on noise.

The algorithm was also tested on real data which was supplied to us by Dr Ayache of INRIA. This data has been extensively used by him and his colleagues for experiments in trinocular stereo matching. Figure 5 shows one of the four triplets which were analysed. Processed line data was also supplied by Dr Ayache based on line detection using a recursive line splitting routine. The characteristics of this line detector are slightly different from that of the Hough based detector. The three images contained 282, 300 and 307 respectively and

our algorithm found 156 correct line correspondences (lines with length less than 10 pixels were excluded). This is better than the 129 line matches found by Ayache's method [2].

Simulated annealing has been adopted in the initial proving stages of our algorithm development as it provides a relatively simple, general purpose and reliable method of finding a global minimum. However it is a computationally intensive method and therefore it is intended to explore more specialised optimisation methods which will better exploit the structure of the problem and lead to faster solutions.

A consequence of the slowness of the optimisation method used was the adoption of heuristic methods to reduce the search space via grouping of all line segments in the same straight line as one feature at an early stage of the process. This potentially introduces a weakness into the method as two lines which physically unrelated but accidentally collinear in one but not all views will be represented by different structures in different images and as the matching scheme enforces unique matches between lines in all frames this may cause some lines to remain unmatched. However, to date we have not found these accidental alignments a common situation.

5 Conclusions

In this paper a novel method of finding correspondences between line segments in trinocular stereo images has been developed. The method exploits constraints based on the orientation of lines and encodes them into a matching function. The method has advantages over previous methods: (1) it works for arbitrary viewing positions whereas previous methods often make the assumption of similarity of viewing positions; (2) it can deal with occlusion and line fragmentation whereas many line based methods fail because calculated endpoints or midpoints of image line segments don't correspond to the corresponding points on the 3D line.

Experimental results show the method is effective for both synthetic and real trinocular image data. Future work will include more detailed studies of the method on a greater variety of images and the development of better computational methods to solve the global optimisation problem.

References

- [1] N. Ayache and B. Faverjon. "Efficient registration of stereo images by matching graph descriptions of edge segments". *International Journal of Computer Vision*, 1(2):107-131, 1987.
- [2] N. Ayache and F. Lustman. "Trinocular stereo vision for robotics". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(1):73-85, January 1991.
- [3] R. Horaud and T. Skordas. "Stereo correspondence through feature grouping and maximal cliques". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(11):1168-1180, November 1989.

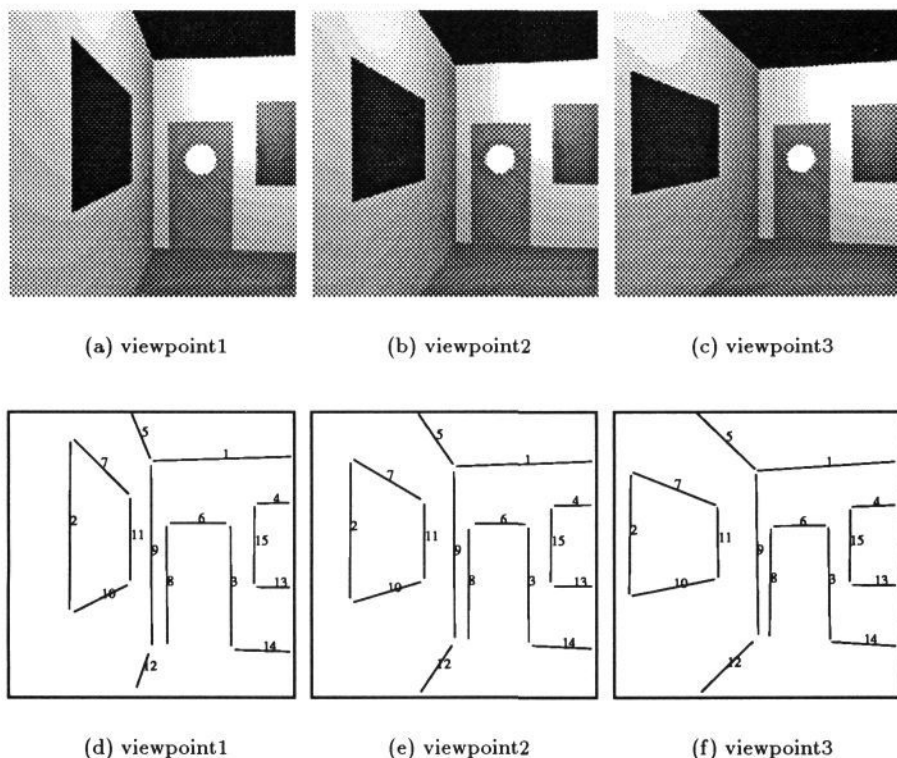
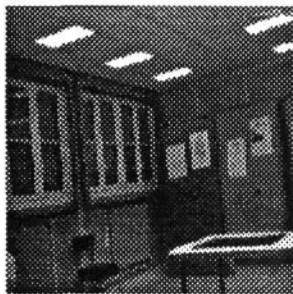
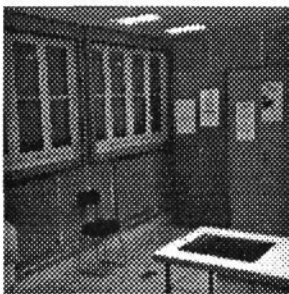


Figure 4: Syntectic Trinocular Stereo Data and Results

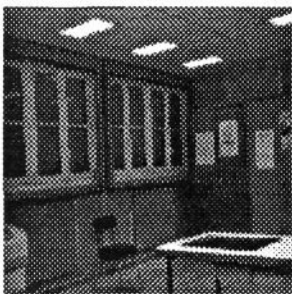
- [4] K. Kanatani. "Computational projective geometry". *CVGIP: Image Understanding*, IU-54(3):333-348, 1991.
- [5] S. Kirkpatrick, C. D. Gellatt, and M. P. Vecchi. "Optimization by simulated annealing". *Science*, 220:671-680, 1983.
- [6] J. H. McIntosh and K. M. Mutch. "Matching straight lines". *Computer Vision, Graphics and Image Processing*, 43:386-408, 1988.
- [7] G. Medioni and R. Nevatia. "Segment-based stereo matching". *Computer Vision, Graphics and Image Processing*, 31:2-18, 1985.
- [8] R. L. Vergnet, S. B. Pollard, and J. E. W. Mayhew. "Stereo-matching of line-segments based on a 3-dimensional heuristic with potential for parallel implementation". In *Proceedings of Alvey Vision Conference*, pages 181-186, 1989.



(a) viewpoint1



(b) viewpoint2



(c) viewpoint3



(d) viewpoint1



(e) viewpoint2



(f) viewpoint3



(g) viewpoint1



(h) viewpoint2



(i) viewpoint3

Figure 5: Real Trinocular Stereo Data and Results