Line Correspondence For Unknown Camera Positions

by

GEOFFREY EDWARD VANDERKOOY
B.A.Sc., University of Waterloo, 1991

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the
Department of Mechanical Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. G. McLean, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. C. Bradley, Department Member (Dept. of Mechanical Engineering)

[redacted]

Dr. D. Hoffman, Outside Member (Dept. of Computer Science)

[redacted]

Dr. K. O. Niemann, External Examiner (Dept. of Geography)

University of Victoria

Abstract

Supervisor: Dr. G. McLean

The correspondence problem of matching features between images from different positions has not be solved for uncalibrated cameras in arbitrary positions. The goal of feature matching is to determine enough matches so that the camera positions can be determined. A line matching algorithm based on the cross-ratio invariant from projective geometry and the absolute intensities of the line border regions is developed. Practical issues of data reduction, invariant cross-ratio identification and signal-to-noise ratio improvement are discussed. The algorithm is tested on synthetic and real image data. The signal-to-noise ratio for typical images turned out to be much lower than expected. Although insufficient for recovery of camera positions, line matches with very low error are achieved for camera rotations about an object of 45 degrees.

Examiners:

[redacted]

Dr. G. McLean, Supervisor (Dept. of Mechanical Engineering)

[redacted]

Dr. C. Bradley, Department Member (Dept. of Mechanical Engineering)

[redacted]

Dr. D. Hoffman, Outside Member (Dept. of Computer Science)

[redacted]

Dr. K. O. Niemann, External Examiner (Dept. of Geography)

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisor, Dr. Ged McLean, for his invaluable assistance in the preparation of this thesis. Thanks also to John Leung and Jason Szabo for encouragement, technical assistance and discussion. Lastly, many thanks to my wife Karen for putting up with me and being supportive throughout.

1. Introduction

There is often a need to obtain the dimension of an object that is difficult to measure. Difficulties arise when the object is large (a chemical plant), small (a Swiss mechanical watch), or the situation is dangerous (inside a nuclear power plant). Complex shapes can also be difficult to measure manually. An automated measurement system could aid the process. A example of this need comes from [Beliveau90] et al.

> *Currently in the United States, approximately 40% of all construction deals with retrofitting existing facilities. The measurement and cataloging of these existing facilities generally cost approximately 7% - 15% of the cost of the facility. This quickly becomes real money.*

At present, measurements of existing structures, commonly called *as-built* measurements, are created manually with some assistive surveying equipment. Original plans (not necessarily including previous changes) may or may not be available. Even when they are, they may not conform to what was eventually built. It is a common occurrence in the gas industry to re-measure a plant after construction to obtain drawings of the *actual* facility.

For the general problem of measuring objects, a vision-based approach is attractive. Such a system would be non-contact, would handle different scales and would provide many measurements from a single image. Some vision-based methods of measurement exist. The basic approaches are as follows:

- Known Target - Objects of known dimension, called targets, are placed into the scene at key locations where at least one will be visible in all images. The targets are designed to be easy to locate and measure accurately in the image. Targets serve as a reference points for registration between images and as calibration objects. From this information, camera and object positions and measurements can be obtained. However, the target, or targets, must be visible in all images and must be able to be placed in the environment.

- Active and Structured Lighting - A laser beam or other light source is projected onto an object which is observed by a camera. The laser and camera are of a known geometry. Either the orientation of the beam is controlled (active lighting) or instead of a beam, some form of light pattern is projected such as a line or grid (structured lighting). The laser light can be easily identified in the camera image, and the known geometry used to determine depth [Ballard82, Schalkoff92]. In essence, the lighting pattern is the target of the previous method. When rotated relative to the object, this can be used to measure the entire object. A common commercial version of this approach is the laser scanner. Objects must fit within the laser scanner's 'work-cell' which is typically on the order of a cubic meter. Accuracy depends on the control of the scanner or object movement and combining data from different scan passes into a useful model is still under development. Laser scanners are also quite expensive ($5K-$200K).

- Stereo Camera Pair - some experimental systems are being developed that utilize two (or more) cameras in a known geometry which use triangulation to determine the depth of objects from the camera [Schalkoff92, Faugeras93]. This requires fairly expensive equipment in the form of a pair of precision cameras mounted in a known geometry. In essence, the second camera replaces the laser of the previous method and the object itself becomes a more complex and unknown target. Most work along this line has been for guidance of autonomous vehicles, not for accurate measurement. Matching between images must still be accomplished, and there must be reasonable camera disparity for depth information to be recovered reliably, limiting the scale or application for a particular setup [Lowe85].

- Photogrammetry - this is a well-developed method that involves manually locating identical points in multiple images. This series of reference points in essence become the target and absolute distance measure. Originally developed for aerial survey, closer methods, commonly referred to as *close-*

range photogrammetry, have also been developed. This method is accurate, but relies on very precise and expensive photographic and optical equipment and is a skilled and manually intensive task [Wolf83].

Most of these approaches are the subject of active research except for close-range photogrammetry in which there has been little work to automate the process. It is a well-developed body of work that can be applied to our problem.

1.1 Definition

There is a need for a good system of non-contact measurement that is automated, simple and relatively inexpensive. For these reasons, the following goal was established as a research project:

> *Using a single camera which is free to roam, obtain a 3-D model of a man-made monument from an assembly of redundant images of that monument without modifying either the monument or the environment.*

The above statement proposes creating automated close-range photogrammetry on a computer. The single camera keeps the cost and hardware complexity manageable. Allowing the camera to roam eliminates tracking systems to correlate images with their positions. The requirement not to modify the environment removes any limits on size or accessibility of the intended object. As long as the object can be viewed from sufficient angles, a model should be possible. It should be possible to allow this flexibility through intelligent use of the data. This idea was proposed by [Beliveau90] et al. That group proposed an original methodology needed to develop a system that would use images and an interactive CAD system to create a CAD model. The idea proposed here goes beyond an interactive system to a completely automated system.

Such a goal has recently become possible in the recent past from the increasing performance and decreasing price of the personal computer and, more importantly, the digital camera. The personal computer puts enough computational ability in one place that

makes it possible to automate the process of measurement. The digital camera gives the flexibility and portability required for a general system. As well, image acquisition is done in a one-step process that can be modeled and calibrated easily.

1.2 The Photogrammetric Method

The need for feature correspondence between images can be seen clearly by considering the standard camera model — the pin-hole camera. The basic relation between world co-ordinates (X,Y,Z) and image co-ordinates (x,y) is a straight line through the focal point (see Figure 1-1).

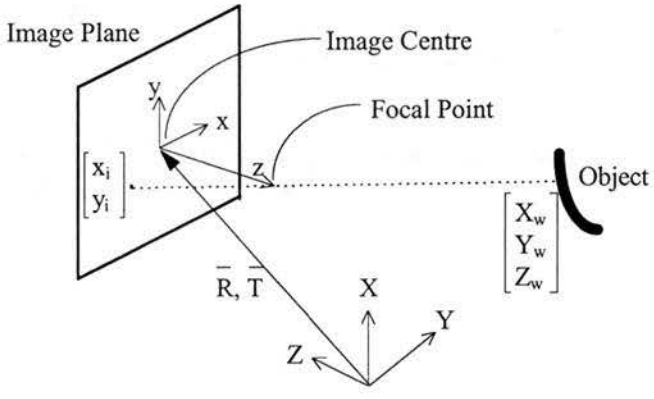

Figure 1-1: Pin-hole Camera Model

The relationship is represented in homogeneous co-ordinates as

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & -1 & f \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where f is the focal length. This can be rewritten as

$$\hat{x}_I = P\hat{x}_W$$

where P is the perspective transformation matrix and w is an arbitrary scaling factor . The relationship is made more complex by the fact that the camera is infrequently aligned with a world co-ordinate system. Three rotations are necessary in general to align the co-ordinates systems. They are

$$R_X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_Y = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_Z = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

As well, there is a translation

$$T = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The image and world units typically differ and must be scaled.

$$S = \begin{bmatrix} S_X & 0 & 0 & 0 \\ 0 & S_Y & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This is simply a series of transformations and can be combined as

$$\hat{x}_I = PTR_X R_Y R_Z S\hat{x}_W$$

The image co-ordinate system is typically located in the upper left corner of the image, and two translations must be done to bring it to the centre of the image, aligned with the focal point. These transformations are usually combined into one general transformation which is represented in homogeneous co-ordinates as

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{13} & a_{24} \\ a_{31} & a_{32} & a_{33} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where w is an arbitrary scaling factor [Gonzalez87]. This is often re-written as

$$[\mathbf{X}_i] = [\mathbf{A}][\mathbf{X}_w]. \qquad [1]$$

The matrix **A** is the *camera calibration matrix*. It encompasses the intrinsic and extrinsic camera parameters. The extrinsic ,or external, camera parameters are the three rotations and three translations which describe the camera position and orientation in world co-ordinates. These change whenever the camera moves. The intrinsic, or internal, camera parameters describe the perspective projection from 3-D to 2-D image co-ordinates and include the focal length, the x- and y-axis unit scaling, and the position of the image centre. These depend only on the geometry of the camera and are not affected by camera movement (although they are affected by focusing changes). These 11 camera parameters

combine to form the general transformation of Equation 1. The calibration matrix is often solved as a unit instead of determining the individual transformations [Schalkoff89].

Other models exist that take into account other aspects of the imaging system. A common extension is to consider the lens distortion from an imperfect lens [Tsai86].

Each 3-D world point maps to an image point via the relationship described in Equation 1. Assuming a camera is in an unknown position, we start with 11 unknowns. Each point relationship results in three equations, three unknowns for the world point positions, and one unknown for the scaling factor. Clearly there are more unknowns than relations no matter how many correspondences are available. However, for an additional camera, we only add the 11 unknowns of the calibration matrix and one for each object scaling factor *assuming the image correspondence is known*. Otherwise, we are no better off. It is through multiple images of the same object and the known image correspondences that the number of equations will outnumber the unknowns and the system be solved.

In general for P points which have been identified in I images taken by C different cameras, the number of unknowns U is defined by

$$U = 5C + 6I + 3P$$

and the number of equations, or constraints R is given by

$$R = 3PI.$$

To solve, we need $R>U$. For different types of correspondences (e.g. lines), the constants differ, but the idea that correspondences across multiple images are required remains the same.

The approach of representing all the unknowns and equations as a large array of relationships and solving is the fundamental concept of bundle-adjustment which forms the

basis of modern photogrammetry [Granshaw80]. This method is well developed and commercial software exists to solve it [Eos92].

However, inter-image correspondences are currently all determined manually. This is the fundamental problem that forms the core of this thesis.

1.3 Feature Matching - Automating Photogrammetry

The main aspect of the project which is not well developed is the stage of feature matching between images, also called the *correspondence problem*. Simply stated, it is to determine which features in an image have corresponding features in another image. There is much work in camera calibration and identifying features in the image - predecessors to feature matching (see Figure 1-2). There is also a substantial body of work on determining camera positions from feature matches: follow-up operations. Work regarding feature matching falls into one of two categories. There is much work in determining correspondence when the camera positions are known (making use of the epipolar constraints) [Faugeras93]. There is also some work for unknown camera positions, but it is for very close images (e.g. video footage) where the images are essentially the same and the problem becomes one of feature tracking between images [Quan88] [Liu91]. There is no method available for the general problem of matching features in images with unknown camera positions.

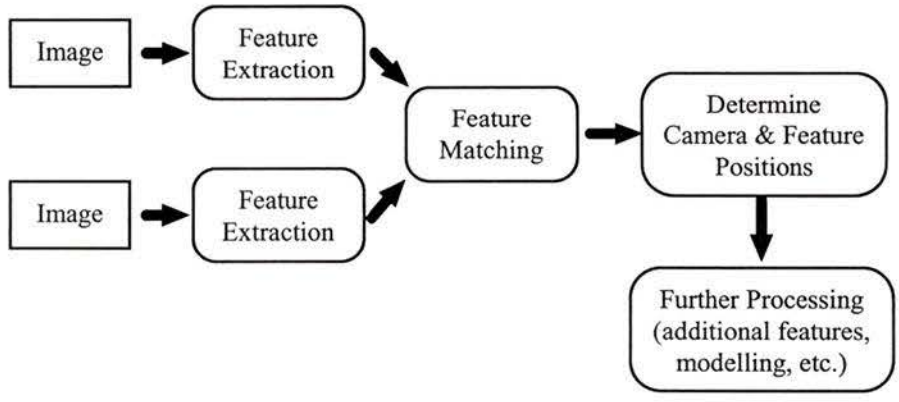

Figure 1-2: Automated Photogrammetric Approach

1.4 The Goal

The goal of this thesis is to be able to determine feature correspondence between images with unknown camera positions. Since there are methods available to determine correspondences if camera positions are known, it is not necessary to determine all correspondences, just sufficient to determine the camera positions. We wish to attempt this by developing a technique that will be minimalist, robust and automated.

Minimalist refers to using a little high quality information and as few assumptions as possible. Much computer vision work uses a hodgepodge of measures, making various and often layered assumptions to achieve a result. These methods are often tuned to a specific environment or set of circumstances. Although assumptions are occasionally required, fewer is better. As will be discussed later, only lines in the image are utilized.

Robustness refers to tolerance to error and generality. Ideally, the technique should be as unlimited as possible from the conditions or assumptions of the environment. Missing conditions and assumptions are viewed by many systems as sources of error and produce erroneous results. Assumptions also limit the generality of a system. We seek a robust system by utilizing only features that are invariant to movements of the camera. In particular, we will use the cross-ratio from projective geometry.

Lastly, the system should be automated. The ideal is that the method is given a task and no user interaction is required to complete the task. As well, as little knowledge as possible should be required from the user in setting up the problem. To this end, the user will only have to provide an image sequence and take pictures with "reasonable overlap".

1.5 Overview of the Thesis

This thesis investigates a novel method of finding line matches in image pairs using the cross-ratio projective invariant, the goal of which is to have enough matches to accurately determine the original camera positions. The use of the cross-ratio would allow it to be

the first of its kind to match lines from disparate views without knowing the relative camera positions (external calibration).

Chapter 2 will present an overview of the features available for matching. Our choice of the line and vanishing point as the basic features will be explained. Attributes of the features will then be considered, concluding that the cross-ratio from projective geometry and the absolute line edge intensity differences are the most robust attributes to use.

Chapter 3 will present our specific problem in detail and develop an algorithm to obtain line matches.

Chapter 4 looks at the implementation issues that must be dealt with, including error and algorithm and data complexity.

Chapter 5 gives the results of a series of tests of the algorithm. Both synthetic and real data sets are used.

Finally, conclusions and ideas for future work will be presented in Chapter 6.

2. Features and Attributes for Matching

The first step in matching is choosing a common and robust feature which can be located in all images and selecting invariant attributes of the feature. This chapter describes the features available for matching and explains the choice of the line as the image feature chosen. The method of finding lines and the resulting attributes is then described. The choice of the cross-ratio from projective geometry is explained, as well as the absolute line border intensity as a secondary attribute.

2.1 Available Features (or Feature Choice)

The main issues in choosing a feature are availability and measurability. Availability signifies that the feature will occur naturally in images and can be identified. Measurability refers to the ability to locate and quantify the feature accurately and reliably. Features must be robust to noise and, ideally, partial occlusion by other objects.

For this project, images are of man-made monuments (e.g. buildings). It is assumed that the lighting is constant and that the scene is static. This is reasonable for non-mobile monuments and only restricts photos to be taken under consistent lighting conditions and over a short time interval. Allowing the lighting to change would result in movement of the shadows (potential features) which is desirable to avoid.

There are many features from which to choose that may be used for matching: region patches, points, straight lines, ellipses & curves. Each of these will now be explored.

2.1.1 Region Patches

Region patches use intensity-based correlations of the pixels in the neighbourhood of a pixel of interest. Pixels of interest are usually chosen for being some form of local extrema (an edge for example). This was one of the original features used in stereo vision algorithms [Dhond89]. Region patches are robust in that they are a distributed feature and therefore reasonably immune to pixel noise (with larger regions being more

robust). They are also attribute-rich since effectively every pixel in the region is an attribute. However, region patches are sensitive to distortion from different viewing positions and the presence of occluding boundaries in the region. They are also highly susceptible to changes in intensity, contrast and illumination, although this is less of an issue for this thesis [Dhond89]. Most importantly, comparisons are intensive to compute involving each pixel in each region. As well, the exact choice of region properties to use for matching is difficult and often dependent on the region itself. This work suffers from the same issues that make texture analysis and matching extremely difficult for a general texture. For these reasons, this feature tends to be little used for general matching, although it is frequently used for video tracking when there is little movement between images. [Faugeras93]

2.1.2 Points

Using only a specific pixel is in essence a region patch with an area of one. Thus, matching will be computationally less intensive, but the pixels are highly sensitive to noise. The location of an edge pixel can easily shift a few pixels. For this reason, specific pixels tend not to be used for generalized matching [Dhond89].

However, these features are used when epipolar constraints are available and searches are restricted to one dimension. In this case, edge pixels (often called *edgels*) are located along an epipolar line, and a match is searched for along the corresponding epipolar line in the other image. This has proven quite practical, but only when the search is restricted to a single line. Note that exploiting the epipolar constraint requires precise knowledge of all camera parameters [Faugeras93].

2.1.3 Straight Lines

Straight lines are common in any environment affected by humans. They are easy to find by locating regions of pixels with similar gradient orientation [Hough62, Burns86]. Since lines are based on a region of edge pixels, they are robust to local pixel noise, well-defined and locatable to sub-pixel accuracy [Burns86, Hussien93].

Data are also reduced by representing all the pixels in the regions simply as a line description and related attributes such as position, orientation and average intensity. The main sensitivity of lines is the extraction of their end-points, and therefore length, which are hard to locate exactly. Interaction with other features causes the end-point positions to be unstable, as well problems with occlusion by other image features [Weng92].

2.1.4 Ellipses & Curves

Like lines, ellipses and curves are robust and well defined [Forsyth90]. Ellipses in particular have the advantage of not having end-points. The problem is that this feature is not necessarily present in all images of interest. Although many man-made monuments involve at least circles, there is no general expectation that a particular type of curve will exist. For this reason, this feature is not a good general choice.

There is a general trade-off between the frequency of occurrence of a feature and the utility. Points and regions are not as robust as ellipses and lines. But ellipses are not guaranteed to be in all images of interest. Given the above, lines were chosen as the fundamental feature for matching. They are a good compromise between a robust, well-supported feature, and one that is likely to be present in any image of interest.

2.2 Line Attributes

The matching process attempts to match features by their properties or attributes. Thus, attributes themselves should be robust to noise and well-defined. In particular, the chosen attributes should be invariant to whatever changes occur between images. Since changes in camera position and orientation are expected, it would be ideal if the attributes were also invariant to illumination level and position changes.

For this project, the Burns line finder was chosen for locating lines [Burns86]. It is capable of finding lines of effectively any size, orientation or intensity. It uses primarily the gradient orientations present in the image to locate lines. Burns et al. argue that the

gradient orientation will vary only slightly along the length of a straight line, whereas the gradient magnitude will vary due to variation of illumination, texture, and noise on either side of the edge.

The Burns method works by determining the gradient orientation at each pixel and grouping pixels with similar orientation into line support regions. Lines can be extracted from line support regions by fitting an intensity plane to the pixels in the region and intersecting with a plane representing the average intensity of the region. Attributes of the line (e.g. intensity, straightness) can then be extracted from the pixels and shape of the line support region [Burns86].

The following attributes are available for each line after line finding:

- **ideal (parametric) line equation** – an R-θ representation is used. This is one where x,y co-ordinates of the line are those that satisfy the equation

$$x\cos(\theta) + y\sin(\theta) = R.$$

The physical interpretation is that R is the length of a perpendicular drawn from the line to the origin and θ is the angle the perpendicular makes with the x-axis (see Figure 2-1).

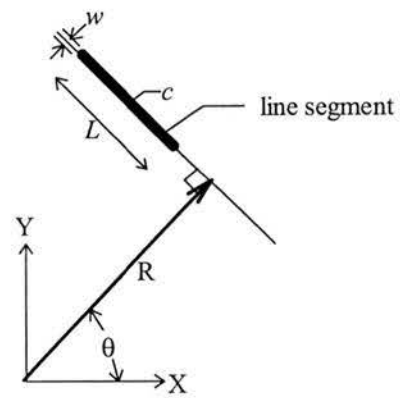

Figure 2-1: Line Representation

Burns determines the line equation using a least-squares fit to the feature values, weighted by the gradient magnitude. Our implementation utilizes a principal components analysis [McLean96].

- **line length** – this is defined as the length of the line that is contained within the line support region and is defined in pixels. The line length is identified in Figure 2-1 by the L. It is approximated for this work by the length of the diagonal of the bounding box for the line support region (see Figure 2-2).

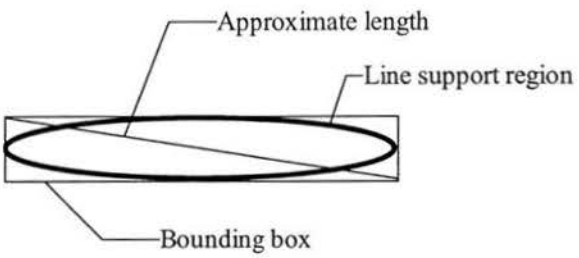

Figure 2-2: Line Length Approximation

- **centroid location** – the x,y location of the centre of the line within the line support region. The centroid is identified in Figure 2-1 by the c. Our implementation locates it during the principal component analysis, essentially as a calculation of the moment of the region [McLean96].

- **average absolute intensity** of the bordering regions – for each side of the line support region, the average pixel intensity value is calculated to determine the average absolute intensity.

- **width** – calculated from the average width of the edge support region and given in pixels. The line width is identified in Figure 2-1 by the w.

The following features may be derived from the above and represent common re-interpretations:

- **contrast** – calculated from the difference in absolute intensity of the bordering regions

- **steepness** – can be defined as the slope of the plane fit to the line support region, or may be derived as *contrast / width*. Alternately, steepness may be defined as the slope of the plane fit to the line support region.
- **region size** – a count of the number of pixels in the line support region. Region size can also be approximated with the line *length * width*.
- **line blobness** - defined as the ratio of length to width.
- **end-point positions** – may be determined from the centroid position, line length and orientation.

A combination of the above attributes may be used to create properties that will be used in the matching of lines.

2.3 Properties of Lines under Perspective Projection

The goal in choosing properties to use in matching is to locate the ones that will be invariant to changes in the perspective projection caused by camera movement. In general, many properties are dependent on the camera position as is implied by the pinhole camera model. The projection process is inherently dependent on the camera position and orientation. Also note that in general, properties associated with line segments (not the ideal line equation) will not be invariant if the segment crosses (or contacts) the image plane boundary. This implies that part of the segment is not visible, thus making region measurements incomplete.

As well as the attributes discussed above, there are also relationships between objects that may be invariant under perspective projection. Such relationships include segment separation and the cross-ratio of four lines. This section will consider all the attributes and relations between lines with regards to their invariance to camera movement. First the previously defined line attributes are considered. Note that for many properties, there are special conditions when the property will not change, but this does not qualify it to be a general invariant.

2.3.1 Ideal Line Equation

The ideal line equation is not generally invariant to camera movement. This can be seen easily from the pin-hole camera model in Chapter 1. Any camera movement changes the A matrix which changes the projection mapping.

2.3.2 Position of segment (centroid)

This property is variant to camera movement for the same reasons as the ideal line equation.

2.3.3 Length

The line length is variant to camera movement. As well, obscuring features may change the line length. It is interesting to note that a line's angular length is invariant to camera rotation.

2.3.4 Absolute Intensity of bordering regions

For many surfaces, it is reasonable to assume that the surface is Lambertian and acts as an ideal diffuser of light [Schalkoff89]. In these cases, the intensity of the regions on either side of the line will be invariant to camera movements. This is obviously not the case for glass and highly polished surfaces. This property is variant to illumination changes.

2.3.5 Line Width

This property is related to the physical geometry forming the line and the imaging system. Depending on the physical geometry, the line width may change with viewing angle. For this reason, this property is variant. It should also be roughly equal for all lines, so this is unlikely to be a good method of differentiating lines

Now the relationships between lines are considered to see if they are invariant. For two lines, the relationships that are available are the intersection point, enclosed angle, distance between the centroids or end-points (measures of separation), and relative length. For any

number of lines, the line order may be considered. For exactly 4 lines there is the cross-ratio. The nature of these relationships to camera movement is now considered.

2.3.6 Intersection point

In general, the intersection point of two image lines is a virtual point not corresponding to any real position and is variant. There are two special cases: corners and vanishing points.

If two lines actually intersect in 3-space, then this intersection corresponds to a real (physical) or virtual corner. This corner has a position in real space and can be treated as a point. It is variant to camera movement but could be used in relationship calculations. However, there is in general no way of determining if a line intersection on the image plane corresponds to the intersection of two lines in 3-space.

If the lines are parallel in 3-space, their intersection will correspond to a vanishing point. Vanishing points correspond to directions in 3-space and are invariant to camera translation. They are variant to camera rotation, although the angles between vanishing points is invariant [Barnard83, Shigang90].

2.3.7 Enclosed Angle

The enclosed angle between two lines (see Figure 2-3) is related to positions of the line segments and is variant to different camera positions.

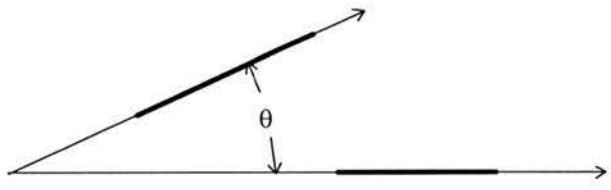

Figure 2-3: Enclosed Angle Between Lines

2.3.8 Separation

Line separation can be measured in many ways - centroid separation or closest end-point separation to name but two (Figure 2-4). However, any measure of line separation is dependent on line position and therefore not invariant.

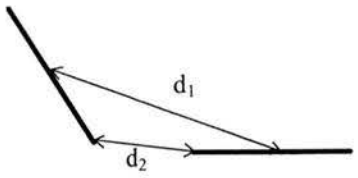

Figure 2-4: Line Separation Measures

2.3.9 Relative Length

Since the line lengths are variant, any ratio of line lengths is variant (Figure 2-5).

Figure 2-5: Relative Length

2.3.10 Line Order

Lines can be ordered by their orientation or by their left-right or top-bottom position (Figure 2-6). The order of lines is variant in general although lines on a plane should keep their order.

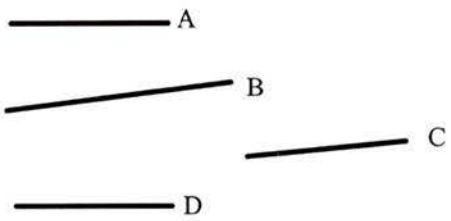

Figure 2-6: Line Order

2.3.11 Cross-ratio

This property comes from projective geometry and is by definition invariant under perspective projection [Semple52]. It is equivalently defined as four lines that cross a single line, or a four lines on a plane that pass through a single point. It will be discussed in detail in section 2.4.

Of the basic line attributes, only the absolute intensity of bordering regions is invariant to general camera movement, and then only when the surface is Lambertian. Of the line relations available, vanishing points are invariant to translation, line order is invariant when the lines are restricted to a plane and the cross-ratio is a general invariant.

The cross-ratio was chosen as the principle property to use in line matching. It is minimalist since it requires only the parametric equations of the lines for its calculation. It is also robust since it is invariant to any camera movement. Line order is implicitly used by the cross-ratio as will be explained when the cross-ratio is considered in more depth.

Vanishing points were chosen to be a line-matching preprocessing step, being undertaken in another thesis [Leung96]. Vanishing points are matched between images using the invariant property of the angles between them. This breaks the lines of each image down into matched groups (perceptual grouping [Lowe85]). The lines from a vanishing point can only match lines in another image corresponding to the same vanishing point. This work also determines the relative camera rotation. For the purpose of this thesis, the work of vanishing point matching is assumed to be complete and available.

The absolute intensities of regions bordering lines is chosen as a secondary property, used to confirm or differentiate matches arrived at using the cross-ratio. This property is not a general invariant and the actual variance will be dependent by each surface in an image. However, this property can prove useful in removing all questionable matches and only keeping those that appear to be correspond to Lambertian surfaces.

The cross-ratio and vanishing points are now explained in detail.

2.4 The Cross-ratio

The cross-ratio is a common definition from projective geometry [Semple52, Barrett91]. The cross-ratio is a ratio of lengths between 4 collinear points - A, B, C and D (Figure 2-7).

Figure 2-7: Four Collinear Points on a Line

The notation for the cross-ratio of four points, A, B, C, and D is [A,B;C,D] and is defined as:

$$[A,B;C,D] = \frac{|AC|}{|AD|} \times \frac{|BD|}{|BC|}$$

It is sometimes defined in terms of four lines crossing a single line [Faugeras93], or as four lines in a plane that pass through a single point [Barrett91]. All are subsets of the above definition.

For this work, only the case of 4 coplanar lines that pass through a single point need be considered. This corresponds to 4 parallel lines on a surface that share a common vanishing point. It is clear that one could draw a single line through the four lines, the intersection defining four points, and the cross-ratio being constant wherever the line was drawn from similar triangles (Figure 2-8).

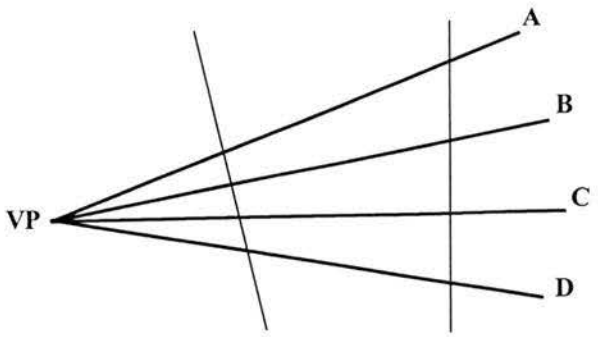

Figure 2-8: Cross-ratio of Four Lines Intersecting a Single Point

The invariant can then be calculated as a cross-ratio of the sine of the angles between the lines as follows [Mohr95, Barret91]:

$$[A, B; C, D] = \frac{\sin(\angle AC)}{\sin(\angle AD)} \times \frac{\sin(\angle BD)}{\sin(\angle BC)}$$

This value will be identical from any viewpoint, not accounting for errors in the process. The reliability of this invariant and others has been considered by [Coelho92] et al. They discovered that the cross-ratio gave them repeatable results around the correct values, ±2%.

Note that the cross-ratio takes four lines and represents them as a single number. The convention for graphically depicting a cross-ratio calculation will be to show it as a bar on a single axis number line, as shown in Figure 2-9. The number line begins at 1.0 (the smallest cross-ratio possible) and goes up. Multiple cross-ratio calculations can be represented by multiple bars. Cross-ratios from different images can be drawn on a separate number line for each image.

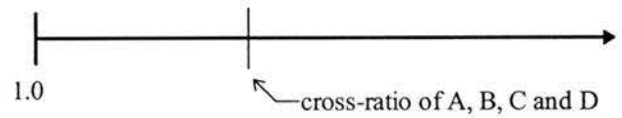

Figure 2-9: Cross-ratio Representation

2.4.1 Cross-ratio Range

The value of the cross-ratio is a real number greater than or equal to one. This can be shown from the definition of the cross-ratio. Consider the basic definition of the cross ratio – a ratio of lengths.

$$[A,B;C,D] = \frac{|AC|}{|AD|} \times \frac{|BD|}{|BC|}$$

Consider the two ratios: AC to AD, and BD to BC. AD is always at least as long as AC, resulting in a value less than 1. As well, BD is always at least as long as BC, the ratio resulting in a value always greater than 1. Since the second ratio is effectively the inverse of the first ratio with the length AB removed from both sides, the second ratio should always be the larger of the two resulting in a final cross-ratio value greater than or equal to one. Coelho et al show that the values will be larger when two lines are close together [Coelho92]. Four evenly spaced lines will give a cross-ratio close to 1. Unevenly spaced lines will have higher values.

2.4.2 Cross-Range Distribution

An important aspect of the cross-ratio is the number of cross-ratios that can be calculated for N lines. This is important because it will directly impact the complexity of any algorithm developed. Assuming that all combinations are valid, there are "N choose 4" combinations which gives the number of cross-ratios I to be given by the formula

$$I = \frac{N!}{(N-4)!4!}.$$

This says that the number of cross-ratios is related to N^4 - an exponential relation. To see the impact of this, see Table 2-1.

Number of Lines	4	5	10	20	50	100	200
Number of Cross-ratios	1	4	210	4845	230,300	3,921,225	64,684,950

Table 2-1: Table of Selected Numbers of Lines and Resulting Cross-ratios

This suggests that some form of data reduction will be required to make the problem manageable on the available computer hardware. As well, unless all the lines are coplanar, then only a small fraction will correspond to valid cross-ratios. Data reduction and invalid cross-ratios will be covered in the discussion of the matching algorithm.

2.5 Vanishing Points and Lines

It is assumed that vanishing points have been found and matched. This has been undertaken by John Leung [Leung96]. The relevant aspects of vanishing points are now presented. A more complete description can be found in [Barnard83].

2.5.1 Vanishing Point Definition

Any set of parallel lines in 3-space, when projected onto an image plane, will appear to converge at a single point called the vanishing point. Vanishing points are a property of the 3-D scene. They are the point at infinity along a line. Since vanishing points are properties of the scene, multiple images of the same scene will contain the same vanishing points.

This phenomena is caused by perspective projection of a 3-dimensional object onto a 2-dimensional image plane. Consider the pinhole camera model as shown in Figure 2-10. A line in space is projected onto the image plane through the focal point. As the line in space (*ls*) approaches infinity, the projection line (*li*) will approach being parallel to line *ls*. They will become parallel when *li* reaches the vanishing point.

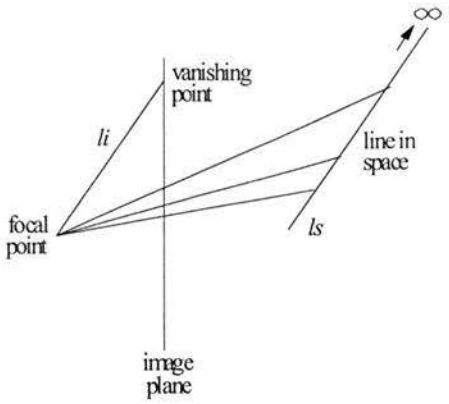

Figure 2-10: Vanishing Point Construction

2.5.2 Lines and Vanishing Point Disappearance

As the camera rotates, the lines that construct the vanishing point may become occluded by the object itself. Sometimes, new lines may appear as a new object face is exposed. Therefore, the number of lines associated with each vanishing point may change drastically from image to image. Also, in some cases when all the lines that construct a particular vanishing point are occluded by the object, the vanishing point itself may disappear between images.

2.5.3 Order of Lines

As camera rotates sideways (or tilted upward/downward), the extruded sections of an object will cause changes in the order of lines. This is shown in Figure 2-11 for both horizontal and vertical lines. Here, the lines **5** and **6** change order, as do lines **a** and **b**. This shows that the order of lines associated with a vanishing point are not invariant. For small camera movements, this will occur infrequently and could be considered invariant, but as the camera motion gets larger, the likelihood of line order changes increases.

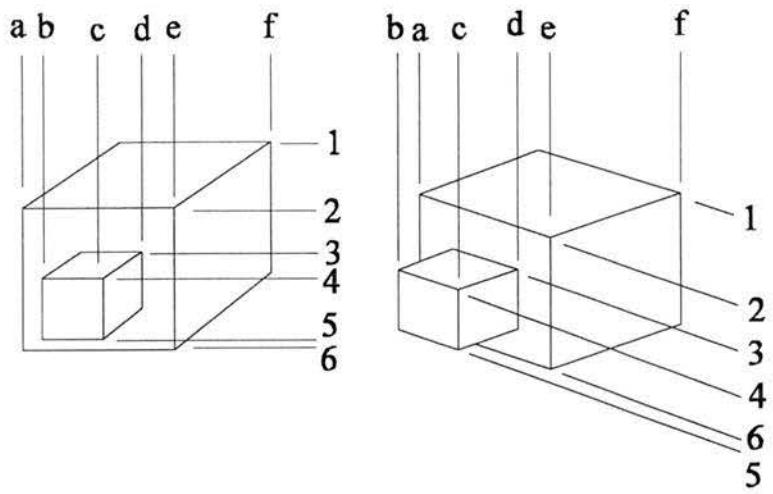

Figure 2-11: Order of Lines for a Vanishing Point

In general, the set of lines that form a vanishing point are shown only to be taken from the set of all parallel lines in 3-space. No assumptions can be made about the presence or absence of particular lines from image to image. As well, the order of lines radiating from a vanishing point is not necessarily consistent, though it may appear to be so for small camera movements.

2.6 Border Intensity

The absolute line border intensity is a secondary feature for proving matches. However, it can be shown that only one of the two bordering regions is guaranteed to be the same in both images.

Consider a line formed by a building corner and the background. As the camera moves, the background changes so the average absolute intensity on that side will vary. However, the intensity of the building side of the edge should be constant assuming that it is not made of glass and reflecting some other area of the environment. So, the intensity difference on the building side may change simply due to random variation. On the other

side, the intensity value may change anywhere on the spectrum depending on what was and is being juxtaposed against the side of the building.

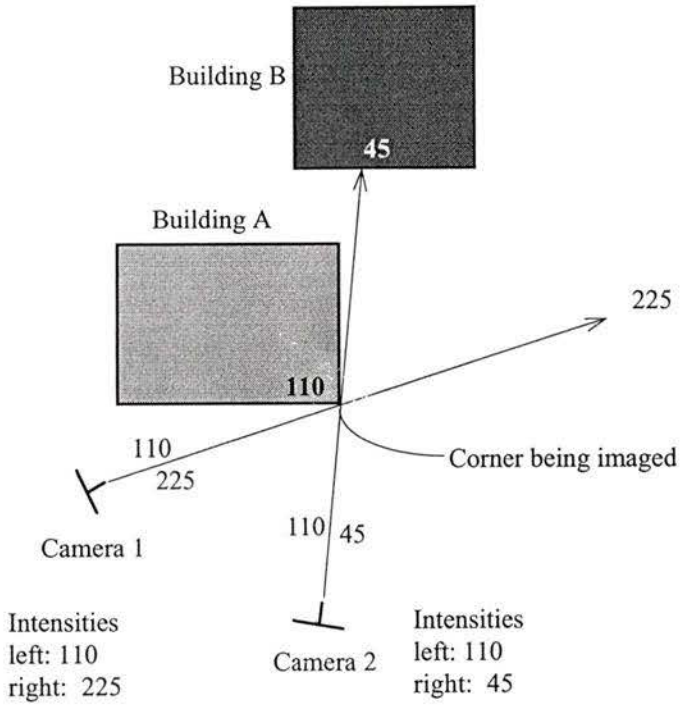

Figure 2-12: Line Border Intensity

In the example shown in Figure 2-12, cameras 1 and 2 are both able to view the lower-right corner of the building and this edge will appear as a line in each image. In camera 1, the right side of the line will be blue sky say. In camera 2, the right side of the image may be dark-red brick from the building B which is now partially visible behind building A. For both images however, the left side of that line will be the colour of building A. Example intensity values are also given in the figure.

For this reason, the differences between both sides of a pair of lines must be considered, and only the smaller difference used to measure similarity.

2.7 Conclusions

In this chapter, various features available for matching have been considered. The line was chosen because it appears in all images of interest, is well-defined and tested methods exist for finding it. The attributes of the line were then considered to see which would be invariant to camera movement and useful for the matching process. Vanishing points were determined to be useful for creating groups of lines that correspond to one another and for determining camera rotation. The cross-ratio was chosen as the primary matching attribute since it is an invariant relationship between four coplanar lines. The absolute intensity of the bordering regions will be used as a secondary property to test tentative matches. The next chapter will develop a matching algorithm using the knowledge of the vanishing points and utilizing the cross-ratio and absolute border intensities.

3. Line Matching

This chapter focuses on the development of an algorithm for line matching using the features and attributes chosen in the previous chapter. First an overview of matching techniques is presented, followed by two relevant techniques that attempted to require only "similar" images. Then the exact goals of the line matching are defined with regards to the follow-up operation of camera position determination. In particular, the number of lines and restrictions will be covered. The choice of cross-ratio definition to use is then discussed. A simple matching algorithm is then presented. The algorithm is progressively developed to the final algorithm which is presented at the end of the chapter. Implementation issues will be discussed in the next chapter.

3.1 Background

Many line matching methods have been proposed. The two major bodies of work, epipolar geometry techniques and tracking techniques, are summarized below to show how they are related but not directly applicable to this work. As stated by Faugeras et al:

> *The problem of matching tokens between images has not been studied widely in the case of motion where the positions of the cameras are unknown. On the other hand it has been fairly well studied in the work on stereo where these positions are assumed to be known.... We do not address this problem in this article.* [Faugeras87]

3.1.1 Epipolar geometry techniques

Epipolar geometry techniques all revolve around knowledge of the camera positions, typically for a stereo camera pair (or triple) whose geometry is known. The methods are based on what Faugeras calls the *epipolar constraint* [Faugeras93]. Essentially this states that for an image point, the corresponding real world point must lie on the line defined by the image point and the focal point. This line in space forms a line on the image plane of any other camera and is called an epipolar line. The image of the same real world point must lie on this line on the other image planes if it exists at all (see Figure 3-1). The

definition of an epipolar plane is the plane defined by the object point and the two focal points. The epipolar lines are then simply defined by the intersection of the image plane with the epipolar plane [Dhond89, Mohr95].

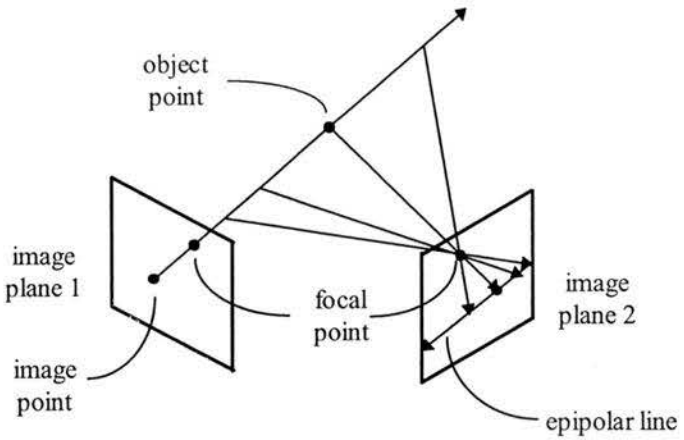

Figure 3-1: Epipolar Line

The epipolar constraint allows for any object located in one image to be searched for along a single line in other images. In essence, it has transformed a two-dimensional search into a one dimensional search. Note that once an object has been found in two images, its exact position can be predicted in a third image as the intersection point of the two epipolar lines. As well, a pair of images can be transformed so that the corresponding epipolar lines lie horizontally in each resulting image plane, to further simplify processing [Mohr95].

These methods are not applicable for this thesis since they require camera positions which are not available, but they will be instrumental in later portions of the project once the camera positions have been located. A good overview of such techniques can be found in [Dhond89] or [Faugeras93].

3.1.2 Video (tracking) techniques

When the images are coming from video, an entirely different method can be used. In these cases, the images from frame to frame are usually very similar, assuming that either the feature of interest or the camera is moving.

Features can be identified in one image, and then located in the next image knowing that they will be in the same vicinity. In essence, the features (or *tokens* as they are often called) are *tracked* from image to image. The search process is simplified from looking throughout the whole image to searching in a very local region. When a feature has been tracked in two images, its position can then be predicted in the next image.

The change in size of an object or feature can be used to determine its speed perpendicular to the camera, and the movement of the centroid parallel to the camera. These changes are often related to a point called the *focus of expansion*: the perceived image point from which the features appear to be expanding. The speeds can be used to calculate position change. Then, two arbitrary images can be used, their relative positions now known, and features located by triangulation, or new features located and placed using epipolar geometry techniques. Relevant work can be found in [Faugeras93] and [Jezouin90].

Unfortunately, for our stated goal, the images are not guaranteed to be similar enough to use this methodology. It should be noted that the overall project could be modified to use video footage, but this would involve a large amount of computation for the increased number of images (frames).

3.1.3 Liu & Huang

Liu and Huang [Liu91] proposed a method of feature matching which is related to work attempted here. In their method, they use lines as their fundamental features for roughly the same reasons already shown. Their method is based on a two-step matching process, and it is stated that the camera movement cannot be too large relative to the object being viewed. The first step is to match a kernel of three lines between images. Next, all other lines are matched relative to these three.

The initial lines to form the *kernel match* are chosen with the following features: 1) long, 2) not parallel, 3) as separated as possible. A small set is chosen from the first image, and a larger set chosen from the second to look for matches. They used 3 lines and 32 lines respectively. The best matches are chosen by looking for the pairs with the smallest matching function: a weighted combination of the disparities in position, orientation, length, width, intensity, contrast, steepness and straightness. Matches which have a matching function value above a threshold are considered failures.

The second step assumes that a kernel match of three lines has been found. The remaining lines are now matched from longest to shortest in the two images, relative to the kernel set. So two lines must have similar differences to the kernel set to be matched.

They demonstrate three test cases, the most impressive of which involves a house where the camera rotated about 15 degrees, keeping at a constant distance and centering the house in the frame. This method is interesting in that it does not require the camera positions to be known a priori, but they must be similar enough that the image appears to be the same. This derives from their assumption that the line properties will be similar, including line position and orientation. Their method was followed up in at least one paper by [Hussien93] et al.

3.1.4 Quan & Mohr

Another relevant method is by Quan and Mohr [Quan88] who attempt to match two images from a "well-defined indoor scene". Their method is actually very similar to that proposed for this project, although with a series of assumptions and restrictions. Two important restrictions are as follows: the camera translation is 1) small and 2) restricted to being parallel to the ground. As proposed here, they use lines as their fundamental features. Next, they group those lines into the following categories: directional groups (by vanishing point), rays (lines with similar angle), collinear groups (very similar angle, assumed to be on a single line).

Their method is as follows:

- Determine the vanishing points using a hierarchical Hough transform. They assume the camera has rotated less than 45 degrees and find the closest vanishing point matches between images on the horizontal plane. This is their *directional* group matching and can be used to determine the (limited) camera rotation.

- The lines from matched vanishing points are now considered. Lines are sorted for each vanishing point by angle, relative to the horizon line. Groups of similar angle (the *rays*) are then extracted and matched between images for similar angle. It is assumed that order does not change between rays since the allowed translation is small. This is called RR matching.

- Rays are then superimposed and the closest line (collinear group) matches minimizing their criterion are chosen. Their criterion is the distance between two segments, calculated as the minimum distance between matching end-points. This is called LL matching.

- The actual segments for each collinear group are matched, based on the order from the VP and their local characteristics (length and orientation). Insufficient description is given to clearly understand this step. This is called SS matching.

Their testing consists of a series of example stages for one pair of images which appear to differ very little. No numerical results are given.

The point of note is that they do use the vanishing points to perform matching of line groups, and from there, attempt to make matches within the groups. Unfortunately, their method is only good for small movements. Possibly this could be applied to video footage, and would allow processing of fewer images than the number required for many tracking algorithms. Their method is limited mainly by the assumptions they make (line order, similar enclosed angle) which are not invariant to camera movement. The proposed method using the cross-ratio should allow this technique to be viable for large angle differences.

3.2 *Determining the Number of Required Line Matches*

As discussed in Chapter 1, six lines across three photos are sufficient to completely define the camera positions. Fewer lines can probably be used if the camera orientations from vanishing point matching are used. That literature is now reviewed to discover what restrictions are on the lines that must be used.

All algorithms for determining camera position from line matches use ideal line representations because line segment lengths are unreliable. It should be clear that the lines cannot all be parallel or there is no reason why any camera cannot slide parallel to these lines. At a minimum, line matches from more than just a single vanishing point will be required. The requirements for the relationship between the lines depend on the type of algorithm used to recovery the camera geometry.

For the non-linear camera position recovery algorithms which were the first to appear, the restrictions are fairly loose. [Petsa94] *et al.* point out that the camera positions are recoverable as long as the following conditions are met: coplanar lines in only two directions with one direction parallel to either a) the plane defined by the focal points, or b) the line defined by two of the principle axes.

Situation a) suggests that if all pictures are taken from the same height, then lines from horizontal vanishing points cannot be used. This is an important restriction, but one that can be easily avoided. Situation b) suggests that the camera should never move directly parallel to one of the vanishing points, or equivalently, should never move parallel to one of the faces of the monument being modeled - another important restriction that can be avoided easily.

For linear algorithms recently developed by the computer vision community, [Weng92] *et. al.* show that the line directions may not be coplanar (Theorem 3) for their algorithm. This in essence requires at least one match for each of three vanishing points or directions. The existence of three orthogonal vanishing points is a reasonable assumption in any man-influenced scene.

These requirements can be visualized in part, as can the requirement for three camera positions. Start by envisioning that each line on the image forms a plane with the focal point in three-space. This plane encompasses the real line, imaged line, and camera focal point (see Figure 3-2).

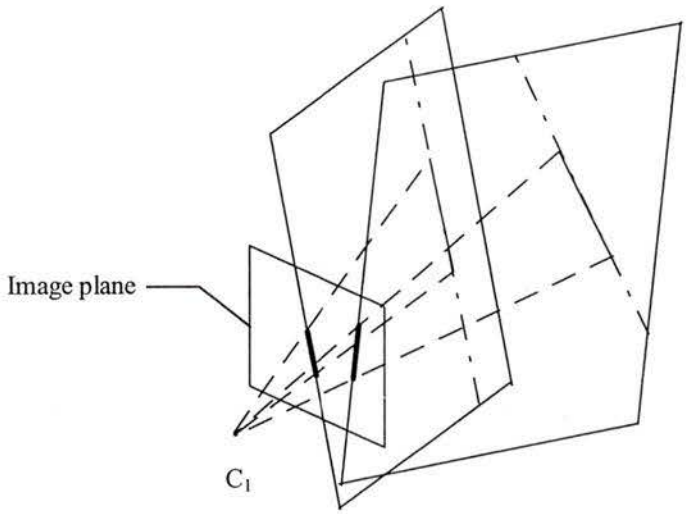

Figure 3-2: Line Planes

Each camera can be considered a point (the focal point) with a set of planes passing through it. For two cameras, wherever planes intersect, this corresponds to the real-world object line. With only two cameras, all matching planes intersect and create plausible real-world placements of lines. There is no constraint on the camera positions – all pairs of planes will intersect to form a line. Two cameras are insufficient for camera placement to be determined (see Figure 3-3). Note that the rays are drawn for a line segment to aid in visualization, but are in fact infinitely long ideal lines.

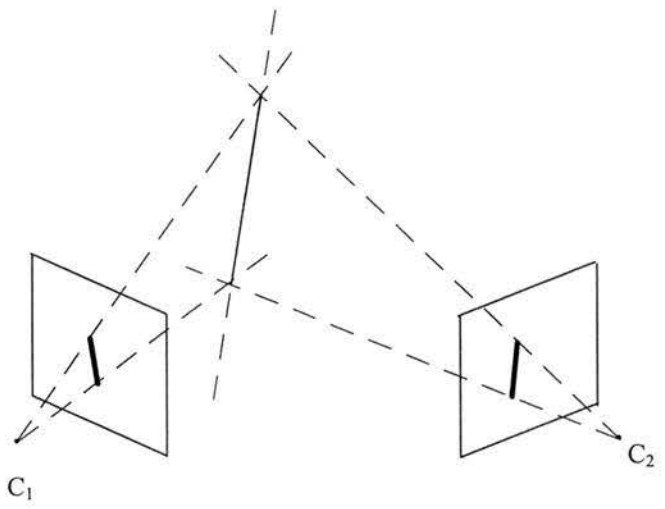

Figure 3-3: Line Planes Intersect at the World Line

Now consider that all the lines in the image are parallel. With three cameras, the third camera must be uniquely placed for three matching planes to intersect in a single line (see Figure 3-4 and Figure 3-5) instead of intersecting at three separate lines (Figure 3-6), for any given placement of the previous two. Thus three cameras actually places a constraint on camera position.

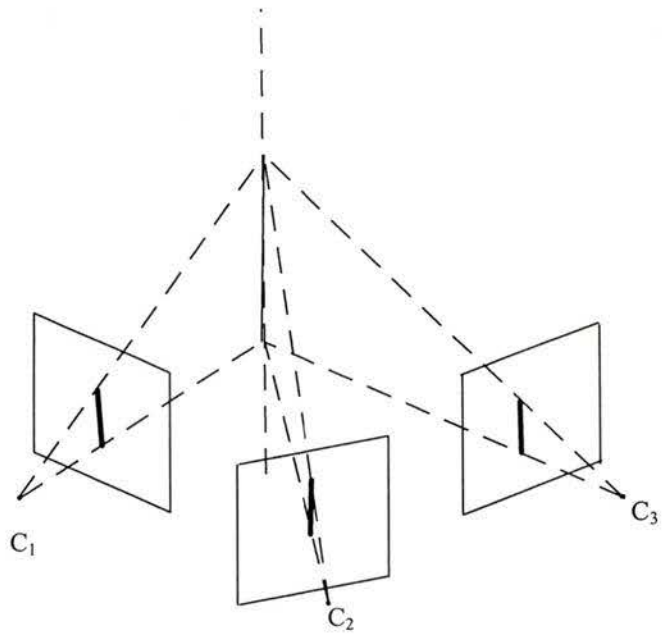

Figure 3-4: Three Line Planes Intersect to Form a Line

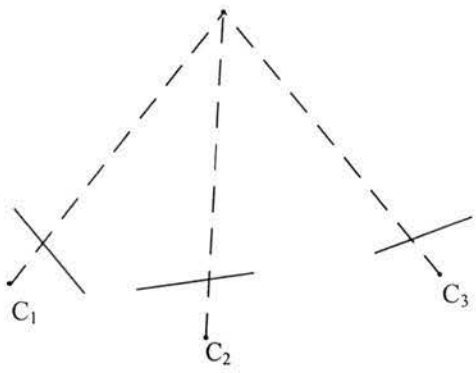

Figure 3-5: End-on View of Figure 3-4 – Three Lines Intersecting at One Point

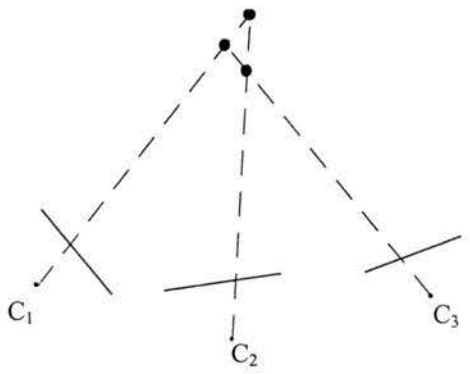

Figure 3-6: End-on View of Three Lines Intersecting at Three Points

Since the lines are infinitely long and assumed for the moment to be parallel in 3-space, then any camera can slide parallel to these lines. The existence of at least one line in another orientation will stop the cameras from sliding relative to one another.

This situation still allows the second camera to be placed arbitrarily with respect to the first. Adding additional lines in the other orientation should make the third camera unplaceable without the first two being in a specific relative locations. This is all determined through an optimization procedure, especially since the system is likely to be overdetermined.

Note that the entire situation can be scaled up or down without reducing generality. The scale of the arrangement cannot be determined without some known measurement. This explanation is not rigorous, but does aid in understanding the restrictions on the lines.

3.3 Calculating the Cross-ratio

There are multiple definitions of the cross-ratio, each implying a different method of calculation. The chosen method will affect the matching algorithm developed. For this application, there is a choice between either four lines with a single crossing line, or four coplanar lines that intersect a common point.

The second definition was chosen since more information that is already known can be utilized, the number of generated cross-ratios will be smaller, and the cross-ratios are more likely to be valid.

For the definition of a common point of intersection, the vanishing point may be used. Then any four lines associated with the vanishing point may be chosen and the corresponding ones from the lines of the matched vanishing point can be searched for. The matched vanishing points have effectively created matched line groups, needing only between-group matches to be found. This method of finding matches within matched vanishing points was also employed by Quan and Mohr [Quan88] as discussed previously. As well, instead of considering every combination of four lines in the image, only the combinations for the lines in a single vanishing point need now be considered. This subdivision of the lines reduces the complexity of the problem and the calculation required.

For the crossing-line definition, every occurrence where four lines appear to touch or cross another line must be located throughout the image. This requires consideration of every combination of five lines in the entire image, which is considerably more calculation than just the sets of four from each vanishing point.

A large concern with both definitions is that of calculating cross-ratios which correspond to the requirements and thus form an invariant structure. This is most easily explained by considering in detail how a cross-ratio is calculated for both methods.

3.3.1.1 Four lines with a crossing lines

For the crossing-line definition, the four lines must intersect the crossing line in 3-space for the cross-ratio calculation to be valid and invariant. Otherwise, the same structure from another angle will have a different cross-ratio value. This situation can easily happen from one object in the image obstructing another. For example, the edge of a building will obstruct the objects behind it. On the image, it will appear that the lines actually terminate at the building edge. Physically these lines do not intersect and the corresponding cross-ratio is invalid. An example is shown in Figure 3-7. Here, the lines for the road and the small building in the background appear to terminate at the edge of the foreground building. An algorithm looking for crossing lines would select these lines and calculate cross-ratios that *does not* correspond to four lines that intersect a single line.

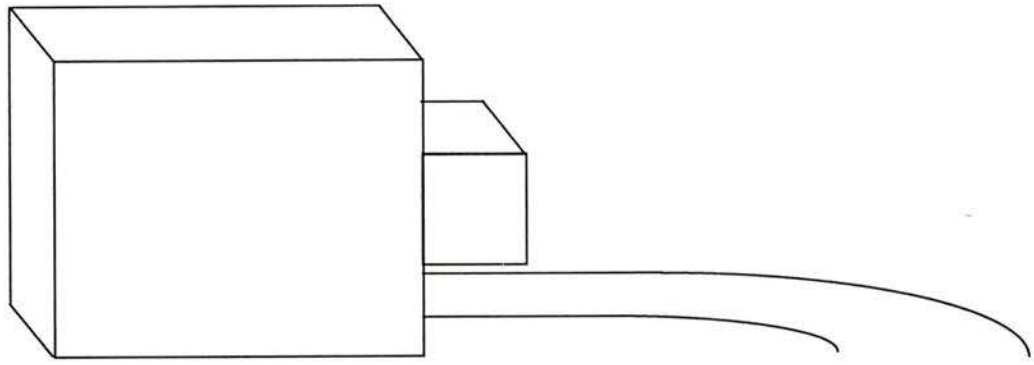

Figure 3-7: Four Lines Intersecting a Crossing Line

Crossing lines, or lines that appear to come close to intersecting are common in images that are being considered. Therefore, invalid cross-ratio calculations are likely to be made. These invalid cross-ratios are noise. Since their values are variant, their matches could appear anywhere on the cross-ratio number line.

3.3.1.2 Four lines and a common intersection point

Although the vanishing points easily supply lines with a common intersection point, the four lines must be co-planar for the cross-ratio calculation to be invariant. Otherwise, the cross-ratio is invalid. There is no known way of proving that any four lines are

coplanar from a single camera image. Thus, if the cross-ratios of all lines associated with a vanishing point are calculated, like the crossing-line method, many of the calculations are invalid, therefore variant and are once again noise. For example, consider the drawing in Figure 3-8, a block house with an open window square. Here, only lines B, C, E and F are coplanar. All other combinations of four lines from A to F are non-coplanar, though they all share the same vanishing point.

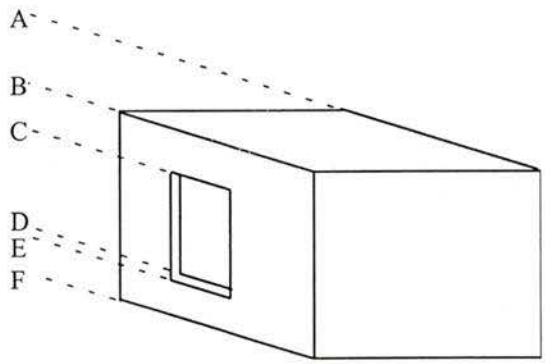

Figure 3-8: Coplanar and Non-coplanar Parallel Lines

For both methods of calculation, there is noise in the form of invalid cross-ratios. Invalid cross-ratios will produce values that will not match the values of the same lines from the other image. Their values are expected to be similar if the camera movement is small, becoming more differentiated as the camera movement becomes larger.

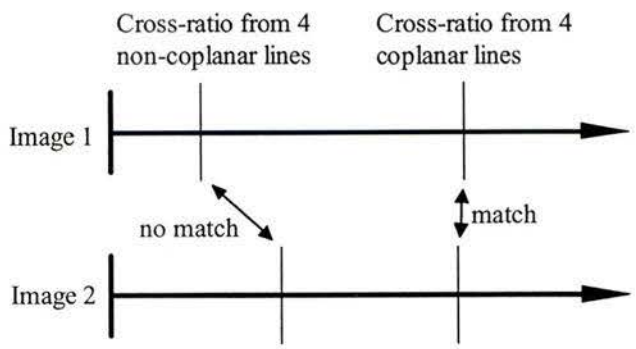

Figure 3-9: Cross-ratio Values for Coplanar and Non-coplanar Lines

It should be noted that valid and invalid cross-ratios will be interspersed equally likely throughout the range of values. There is simply no way of distinguishing them.

3.4 A simple matching algorithm

A simple matching algorithm is now proposed which will be built upon to arrive at the final method. It is assumed that lines have been found in the image, vanishing points have been found from the lines and matched between images. Therefore, the work begins with data for two images:

- VP – a list of the matched vanishing point pairs
- L_{ij} – lists of the lines associated with each vanishing point of a match. i is the image number of the vanishing point, $i \in [1,2]$. There are N_i lines associated with the vanishing point. j is the line index number, $j \in [1,N_i]$.

Since vanishing points are assumed to be found and matched, all algorithms are presented only for a matched vanishing point pair.

In the algorithms developed following, $\angle L_{ij}L_{ik}$ refers to the angle between the jth and kth lines of the vanishing point in the ith image. The lines are assumed to be in order around the vanishing point so that the first and fourth lines are always the outer ones in a set of four.

The value of each cross-ratio calculation is denoted by CR_{iq} where i is the image number, there are C_i cross-ratio values and q is the cross-ratio index, $q \in [1, C_i]$.

Each cross-ratio match, $CR_{1i} \leftrightarrow CR_{2j}$, is denoted by CRM_m, where there are M matches and $m \in [1, M]$. Each cross-ratio match implies four individual line matches, $L_{1l}\, M_m \leftrightarrow L_{2l}\, M_m$ where l is the line match index, $l \in [1, 4]$.

The difference between two cross-ratios, used to determine the best match, is normalized by the average value. This is because [Coelho92] showed that the amount of error was

proportional to the magnitude of the cross-ratio. For simplicity, this difference between cross-ratios will be denoted by ΔCR and is defined by

$$\Delta CR(ni, mj) = \frac{\left| CR_{ni} - CR_{mj} \right|}{(CR_{nk} + CR_{mj})/2}.$$

This algorithm makes use of bi-directional matching, a sensible way to locate the best matches in the data. Consider the situation shown in Figure 3-10.

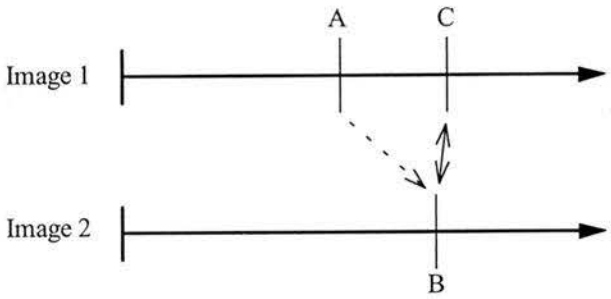

Figure 3-10: Bi-directional Matching

Here are cross-ratios from two images. Looking for matches for the cross-ratios in Image 1, the best match for A would be B. This is the "forward" match. However, if the best match for B is then looked for, it will be found not to be A, but instead C. Therefore, A will not be matched because it was not consistently the best match for both cross-ratios. When C is attempted to be matched, it's best match will be B, whose best match will also be C. B and C will be correctly matched.

This is sensible since not all cross-ratios will necessarily have a matching cross-ratio: one of the lines may only be present in one image. Bi-directional matching is a method of handling cross-ratios without matches.

From this information, the matching algorithm for a pair of matched vanishing points is proposed which utilizes only the cross-ratio information. It is advantageous to use *only* the cross-ratio if possible to keep the required information minimalist. The algorithm has two basic components, which are executed for each vanishing point pair. The first part

calculates all the cross-ratios for the two vanishing points. The second part searches for matches. The algorithm is as follows:

Calculate cross-ratios for image 1

$\forall\ i, j, k, l \in [1, N_1], i < j < k < l$

$$CR_{1q} = \frac{\sin \angle L_{1i}L_{1k}}{\sin \angle L_{1i}L_{1l}} \times \frac{\sin \angle L_{1j}L_{1l}}{\sin \angle L_{1j}L_{1k}}, q = 1,2,\ldots C_1$$

Calculate cross-ratios for image 2

$\forall\ i, j, k, l \in [1, N_2], i < j < k < l$

$$CR_{2r} = \frac{\sin \angle L_{2i}L_{2k}}{\sin \angle L_{2i}L_{2l}} \times \frac{\sin \angle L_{2j}L_{2l}}{\sin \angle L_{2j}L_{2k}}, r = 1,2,\ldots C_2$$

$\forall\ i \in [1, C_1]$

Forward match

$\exists\ j \in [1, C_2]$ that minimizes $\Delta CR(1i, 2j)$

Backward match

$\exists\ k \in [1, C_1]$ that minimizes $\Delta CR(1k, 2j)$

Decision

if $i=k$, register match: $CRM_m = CR_{1i} \leftrightarrow CR_{2j}$, $m = 1,2,\ldots M$

This algorithm is simple, using only the cross-ratio and locating the best match for each cross-ratio, if there is one.

3.5 Reducing the noise

As noted earlier, there will be a number of noise cross-ratios: cross-ratio calculations where the lines used did not meet the requirements for the cross-ratio to be invariant. Methods to reduce the chance of forming a match based on these noise values are now considered: choosing cross-ratio matches that differ only by a reasonable amount of error,

choosing only unique matches, and keeping only consistent results. Each of these options is now discussed.

3.5.1 Cross-ratio Match difference

The main option in the matching process is the normalized difference in cross-ratio value, ΔCR, of a potential match. This difference is normalized by the average value because the variance in the cross-ratio is proportional to the magnitude [Coelho92]. Coelho *et al.* also discovered that cross-ratios differed by up to 2% due to measurement error. It would be sensible to look for matches only within the range of expected error of the cross-ratio. Values farther spread than this must differ for some reason other than measurement error: probably they are invalid cross-ratios. Therefore, a threshold of maximum error for a potential cross-ratio match is defined, the *match threshold*, and denoted by T_M. The match threshold will be set to the maximum expected error in the cross-ratio measurement. This threshold defines a range within which matches are considered called the *match range*. The cross-ratio difference of a potential match must be less than this to be considered for matching. An illustration of the match range is shown in Figure 3-11.

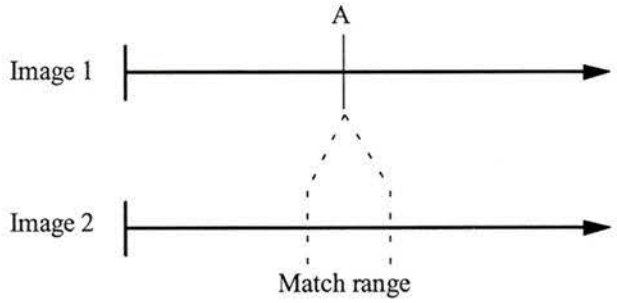

Figure 3-11: Match Range

The matching algorithm is therefore changed to find matches that not only minimize the value of ΔCR, but also require

$$\Delta CR < T_M$$

for a valid match.

3.5.2 Uniqueness

Since the cross-ratio represents a spatial relationship between lines which is scale independent, the same cross-ratio value can be caused by another occurrence of the line structure, or even the same structure at another level of scale. This can result in the situation shown in Figure 3-12 where a similar structure occurs in image 2. Here, the correct match is unclear.

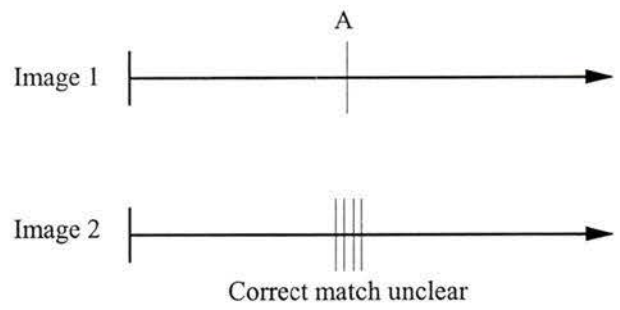

Figure 3-12: Close Cross-ratios from Similar Structures

The task of locating real matches may be simplified by looking only for unique or clear matches - matches that have no close neighbours. By not considering similar values, mis-matching these items is avoided. As well, this will cut down the number of matches that must be considered.

From this idea, the *uniqueness range* is defined within which other cross-ratios must not be present for the cross-ratio to be considered unique. This can be implemented by requiring that the difference between a cross-ratio and its neighbour be greater than the *uniqueness threshold*, denoted by T_U. The difference is defined as a percentage of the cross-ratio value since the variance in the cross-ratio is proportional to the magnitude [Coelho92]. Non-unique cross-ratios will be removed from the list of cross-ratios to be matched. The relationship is defined such that a cross-ratio i is unique if there exists no cross-ratio j which fulfills the relationship

$$\frac{\left|CR_{ni} - CR_{nj}\right|}{CR_{ni}} < T_U .$$

This is similar to the match threshold, except it applies to the cross-ratios from the same image instead of from different images. In Figure 3-13, the cross-ratios A and E would be used for matching, but not B, C or D because they are within each other's uniqueness range.

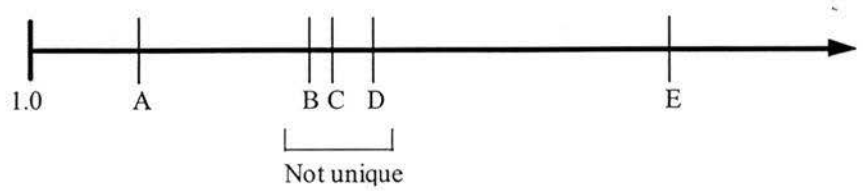

Figure 3-13: Cross-ratio Uniqueness

The algorithm for this uniqueness criteria of a single vanishing point is as follows.

$$\forall\, i \in [1, C]$$

$$\text{if } \exists\, j \ni \frac{\left|CR_i - CR_j\right|}{CR_i} < T_U$$

then remove CR_i and CR_j

By setting the uniqueness range equal to the maximum expected error in cross-ratio calculation (and hence the match range), we ensure that each cross-ratio corresponds to a unique structure in its image. This avoids mis-matches due to repeated structures as well. The maximum expected error is the default setting for the uniqueness range and threshold.

As long as the match range is less than or equal to the uniqueness range, there can be only one match available for any particular cross-ratio. However, if the match range is larger than the uniqueness range, unexpected results can occur. A series of crowded but closer cross-ratios can get removed, leaving a cross-ratio matched with one which is even farther away. Thus, the match range must not be larger than the uniqueness rating of the data. For example in Figure 3-14, A's best match is C. However, B, C and D may get removed because they are not unique, leaving A's best match to be E, an incorrect choice.

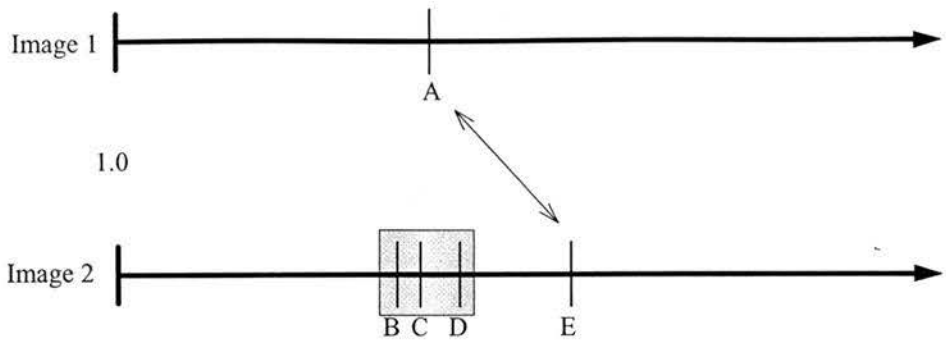

Figure 3-14: Requirement for a "Dead Zone" for Non-unique Cross-ratios

If the uniqueness range is smaller than the matching range, a dead-zone must exist which signifies that cross-ratios did exist there but are not available for matching. For the implementation, the matching range is kept smaller than the uniqueness range. This could be changed if testing revealed that insufficient matches were being made. This also simplifies the implementation. The effects of uniqueness range size will be examined later.

In summary, after cross-ratios are calculated, the list will be culled to remove all non-unique cross-ratios.

3.5.3 Consistency

The cross-ratio represents four lines as a single scalar for matching. Cross-ratio matches then correspond to four line matches. Different cross-ratio matches can involve subsets of the same lines. This can result in inconsistent line match results.

Cross-ratio match 1

Image 1	Image 2
A	M
B	N
C	O
D	P

Cross-ratio match 2

Image 1	Image 2
C	O
D	Q
F	R
H	T

Table 3-1: Inconsistent Line Matches

Consider the example pair of cross-ratio matches shown in Table 3-1. The lines C and D from image 1 occur in both matches, as does the line O from image 2. Notice that line D is matched to both line P and line Q. This is not possible: the line-matches are inconsistent. This mismatch of lines has occurred in part because of the 4-to-1 mapping of lines by the cross-ratio.

This form of cross-ratio mismatch could occur because of:

- noise causing incorrect cross-ratio matches.
- the cross-ratios are invalid and randomly ended up getting matched.
- one or more lines are not present in one of the images so no true match was present and an invalid match occurred instead.

There are a number of ways of dealing with this situation:

- remove the bad line matches
- remove the cross-ratio matches
- keep only the "best" cross-ratio match
- keep the line-matches or cross-ratios with the most "support"

The best cross-ratio match could be defined as the one with the least difference in cross-ratio, implying the least error in the underlying measurements. The line-matches or cross-ratio with the most support could be defined as the one with which the most other cross-ratio match results agree. This is the most complex option and would only be useful with enough cross-ratio matches.

Without exact knowledge as to when inconsistent line matches occurs, the safest approach is to remove both cross-ratios. During testing, if a trend was discovered, a more appropriate response could be developed.

This step of consistency checking would occur after all cross-ratio matches had been determined: a post-processing step. The algorithm is shown below.

For all line matches from all cross-ratio matches

$\forall\, m \in [1, M]$

$\forall\, l \in [1, 4]$

Compare to all line matches from all other cross-ratio matches

$\forall\, n \in [1, M],\ n \neq \mathrm{m}$

$\forall\, k \in [1, 4]$

Look for an inconsistent line match

if $(L_{1l}M_m = L_{1k}M_n$ and $L_{2l}M_m \neq L_{2k}M_n)$

or $(L_{1l}M_m \neq L_{1k}M_n$ and $L_{2l}M_m = L_{2k}M_n)$

then remove CRM_m and CRM_n

3.6 Selecting Cross-ratios

The algorithm developed so far simply determines all cross-ratios and assumes that cross-ratios corresponding to invariant structures will form matches and others will not. In essence, there is signal (valid cross-ratios) and noise (invalid cross-ratios) which can be represented as a signal-to-noise ratio (SNR). The SNR will be dependent on how many planes there are for each vanishing point, and how many lines are on each plane. This section presents a series of heuristics that may be effective in selecting a set of cross-ratios that is not only smaller than the set of all possibilities, but will have a higher signal to noise ratio than that of all possible cross-ratios. This is done instead of calculating all possible cross-ratios.

Since ideally only one cross-ratio match is required for each vanishing point, the set of cross-ratios for matching can be reduced in size significantly as long as there are still valid matches that are identifiable in the data. A heuristic method of selecting cross-ratios that

removed most of them would be acceptable as long it maximized the signal to noise ratio of the remaining values.

A number of methods for selecting cross-ratios from the list of all possibilities are suggested as alternatives to using all the data available. They have been titled as follows:

- End-point alignment selection method
- Equi-radial selection method
- Centroid-line selection method
- Combined end-point and equi-radial selection method

All of these methods are alternatives to doing the complete calculation for all lines associated with each vanishing point. Many of these are inspired by ideas from [Lowe85]. He proposes using perceptual grouping and whatever can be reasonably inferred from the given data. All the methods look for a particular spatial relationship between the lines. The relationships are attempts at inferring coplanarity of the lines. One feature they all share is that they tend to reject widely separated lines as being coplanar. Note that all methods will miss some valid cross-ratios and accept some invalid cross-ratios.

Each of these methods will now be described, as well as the reasoning behind them. Quantitative analysis will be covered with the results of the matching: the true objective.

3.6.1 End-point alignment selection method

The goal here was to calculate a cross-ratio only if the line end-points of one end were "reasonably" aligned, e.g all existed in a band 20 pixels wide (see Figure 3-15). This heuristic was envisioned as a simple approximation to finding a crossing line. Lines that seem to end in a straight line will either correspond to objects being obstructed by another object (not useful) or by texture and objects on a plane that stops at the edge. The later case is hoped to infer coplanar lines. The width of the band is used to compensate for line-length detection inaccuracy.

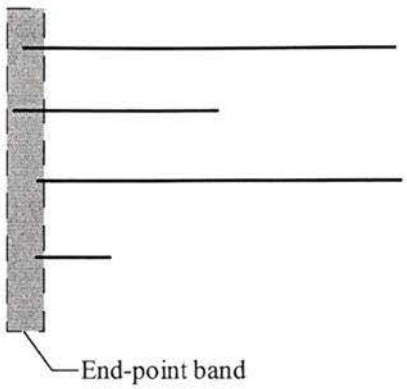

Figure 3-15: End-point Band

For implementation simplicity, the band was defined to exist around the line connecting the end-points of the outer lines. If the inner line end-points are within the separation threshold, T_S, then they are considered aligned. This selection algorithm is as follows.

$\forall\ i, j, k, l \in [1, N],\ i < j < k < l$

$\exists$ lines E_1 and E_2 defined by the endpoints of L_i and L_l on the same side

if ($|E_1$ to L_j end-point$| < T_S$ and $|E_1$ to L_k end-point$| < T_S$)

or ($|E_2$ to L_j end-point$| < T_S$ and $|E_2$ to L_k end-point$| < T_S$)

$$CR_q = \frac{\sin \angle L_i L_k}{\sin \angle L_i L_l} \times \frac{\sin \angle L_j L_l}{\sin \angle L_j L_k}, q = 1,2,\ldots C$$

3.6.2 Equi-radial selection method

This method states that a cross-ratio will be calculated for four lines only if there is an arc of radius R, centered on the vanishing point, that will intersect all four lines. In essence, the four lines must exist at some common distance from the vanishing point (see Figure 3-16).

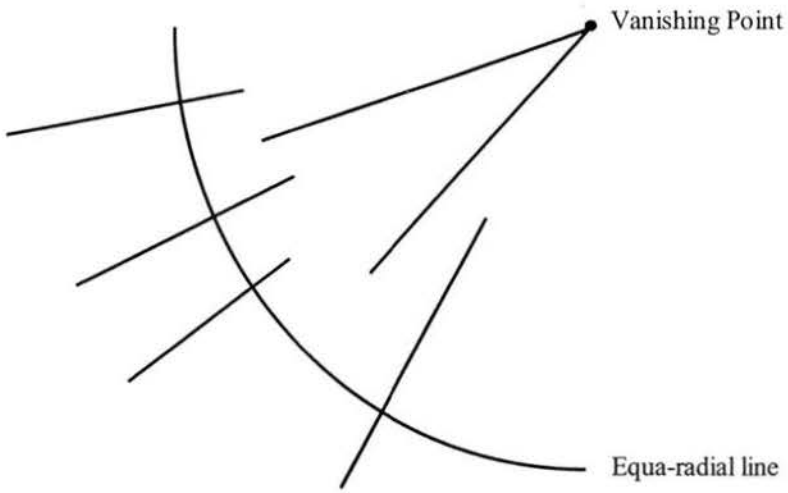

Figure 3-16: Equi-radial Line

This method is intended to have only lines that are side-by-side, which are more likely to be coplanar than lines that are not. It tends to stops line segments from opposite sides of the image with no overlap being used to form a cross-ratio. This method is similar to the previous one without requiring that end-points align. It allows lines that are contained between other lines to be utilized. Figure 3-17 illustrates a situation where this is desirable.

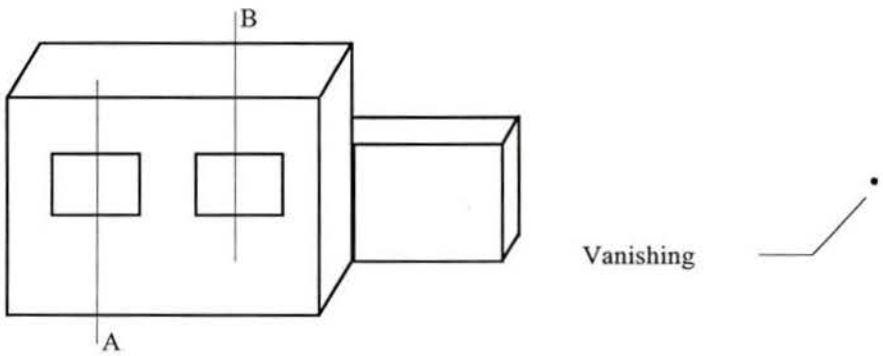

Figure 3-17: Valid Equi-radial Line Illustration

Here, we would like to avoid creating cross-ratios with lines from both the square buildings. This method effectively avoids this mistake. The line **A** crossing the lines from

the top and bottom of the building and the windows makes a valid cross-ratio; however, none of the end-points align. But they do exist at a common distance from the vanishing point (which is drawn more closely than it should for illustrative purposes). Note that this does not guarantee valid cross-ratios, as shown by the line **B**. The algorithm is as follows.

$\forall\ i, j, k, l \in [1, N], i < j < k < l$

if $\exists$ an arc A of radius r, centered on the VP $\ni A$ intersects L_i, L_j, L_k and L_l

$$CR_q = \frac{\sin \angle L_i L_k}{\sin \angle L_i L_l} \times \frac{\sin \angle L_j L_l}{\sin \angle L_j L_k}, q = 1,2,\ldots C$$

This method can also be implemented as a lower-order algorithm, instead of order N^4 to calculate all possible cross-ratios. A list is created of end-points for lines and their distances from the vanishing point (order N). The list is then sorted (order NlogN for a good sort algorithm) and processed in order to keep a running list of which lines are in any range from the vanishing point (order N). Each time a new line is added, all new cross-ratios with that line are calculated.

This method can miss lines that are aligned, but due to viewing angle, are at a slope relative to the vanishing point. For this reason, the following, more general method was developed.

3.6.3 Centroid-line selection method

This is an attempted improvement on the previous method. The goal is to try to efficiently find line sets that are aligned, but not necessarily equi-distant from the VP. An example of this is a line of windows on the side of a tall building observed from the ground. A horizontal line (perpendicular to the vertical vanishing point) cannot always be found that intersects all the windows.

The original intent was to calculate the cross-ratio for any four lines for which there exists a line that can be drawn through all four lines. This removes the restriction of being equi-radial from the vanishing point and still requires aligned structure. For implementation

and search efficiency, this was reduced to determining whether a line connecting the centroids of the outer lines intersected the two inner lines (see Figure 3-18).

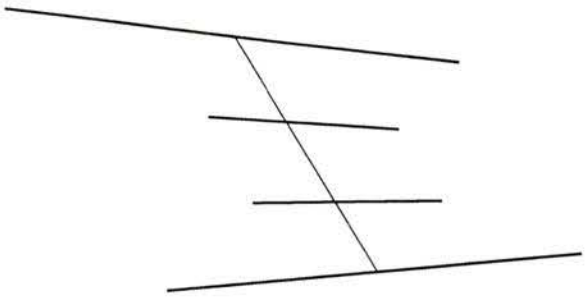

Figure 3-18: Line Alignment Example

The algorithm for this method is as follows.

$\forall\ i, j, k, l \in [1, N],\ i < j < k < l$

$\exists$ line A defined by the centroids of L_i and L_l

if A intersects L_j and L_k

$$CR_q = \frac{\sin \angle L_i L_k}{\sin \angle L_i L_l} \times \frac{\sin \angle L_j L_l}{\sin \angle L_j L_k}, q = 1,2,\ldots C$$

3.6.4 Combined end-point and equi-radial selection method

This method attempts to combine the inferred plane of the end-point alignment technique with the faster calculation of the equi-radial technique. The goal was to target an even smaller subset of lines. This technique may be useful in reducing very large data sets.

Line sets that would be acceptable for the equi-radial method described previously were checked to ensure they also had aligned end-points on at least one side. For these sets of lines, cross-ratios would be calculated. The algorithm is as follows.

$\forall\ i, j, k, l \in [1, N], i < j < k < l$

if $\exists$ an arc A of radius r, centered on the VP $\ni A$ intersects L_i, L_j, L_k and L_l

$\exists$ lines E_1 and E_2 defined by the endpoints of L_i and L_l on the same side

if ($|E_1$ to L_j end-point$| < T_S$ and $|E_1$ to L_k end-point$| < T_S$)

or ($|E_2$ to L_j end-point$| < T_S$ and $|E_2$ to L_k end-point$| < T_S$)

$$CR_q = \frac{\sin \angle L_i L_k}{\sin \angle L_i L_l} \times \frac{\sin \angle L_j L_l}{\sin \angle L_j L_k}, q = 1,2,\ldots C$$

3.7 Using the border intensity

Up to this point, the matching algorithm has been based exclusively on the cross-ratio. Now consider using the line border intensity to verify or improve the matching algorithm. Its fundamental application is to determine whether two lines are similar.

To decide if two lines are similar, the minimum difference in border intensity is used as a measure. This difference must be smaller than the *intensity threshold*, T_I, for the match to be considered valid. The intensity threshold is defined to be a fixed number of intensity levels. The minimum border intensity difference, $\Delta I(i,j)$, between lines i and j is defined as the following, where IL_i and IR_i are the intensity on the left and right sides of line i, respecively:

$$\Delta I(i,j) = \min(|IL_i - IL_j|, |IR_i - IR_j|).$$

This can be used to compare lines in the same images or in different images.

Since many surfaces are not Lambertian, some intensity variation should be expected so the intensity threshold should not be too restrictive. Essentially this is a coarse filter to remove bad matches, so the exact value is not crucial.

The border intensity difference can be used as either a post-processing match verification, or in the matching process to avoid considering invalid matches.

3.7.1 Proving match results with the intensity property

As a post-processing step, the intensity difference would be used before the consistency of the data is checked. For this step, each of the four individual line matches from a cross-ratio match must have intensity differences less than the intensity threshold, and therefore be considered similar. Cross-ratio matches with one or more non-similar line matches would be discarded. This functions as a stand-alone proof of the matches obtained using only the cross-ratio. The algorithm for this is as follows.

For all line matches from all cross-ratio matches

$\forall\ m \in [1, M]$

$\forall\ l \in [1, 4]$

Look for an inconsistent line match

if $\Delta \mathrm{I}\ (L_{1l}M_m, L_{2l}M_m) > T_I$

then remove CRM_m

3.7.2 Using the intensity property in the matching process

Instead of using the intensity process to check matches after they have been created by the cross-ratio algorithm, it makes sense to check them during the matching process. The algorithm could be designed to consider only cross-ratio matches where the lines themselves could be matched. The matching portion of the algorithm is as follows.

$\forall\ i \in [1, C_1]$

if $\exists\ j \in [1, C_2]$ that minimizes $\Delta\mathrm{CR}(1i, 2j)$ and $\Delta\mathrm{CR}(1i, 2j) < T_M$

and $\forall\ m \in [1, 4]$, $\Delta\mathrm{I}(\ 1im, 2jm) < T_I$

then if $\exists\ k \in [1, C_1]$ that minimizes $\Delta\mathrm{CR}(1k, 2j)$ and $\Delta\mathrm{CR}(1k, 2j) < T_M$

and $\forall\ m \in [1, 4]$, $\Delta\mathrm{I}(\ 1km, 2jm) < T_I$

then if $i = k$, register match: $CRM_m = \mathrm{CR_{1i}} \leftrightarrow \mathrm{CR_{2j}}$, $m = 1,2,...M$

Since cross-ratios are now matched only if the lines could be matched, it is sensible to redefine the uniqueness criteria to reflect this. A cross-ratio is unique if there are no other cross-ratios within the uniqueness range *that could be matched to the same cross-ratio in*

the other image. To be matched to the same cross-ratio in the other image, both cross-ratios in the image must have similar lines, as defined by the intensity difference criteria. This replacement for the uniqueness portion of the algorithm is as follows.

$$\forall\, i \in [1, C]$$

$$\text{if } \exists\, j \ni \frac{\left|CR_i - CR_j\right|}{CR_i} < T_U$$

$$\text{and } \forall\, m \in [1, 4],\ \Delta\mathrm{I}(im, jm) < T_I$$

then remove CR_i and CR_j

It is expected that this algorithm will result in more "unique" cross-ratios than without using the intensity property. More are retained since the intensity property allows cross-ratios with similar values to still be considered unique as long as their lines are different. Obviously, the post-processing intensity consistency check is redundant given the above changes to the uniqueness and matching portions of the method.

3.8 The Proposed Algorithm

Given the original simple algorithm that was proposed and the following improvements, essentially three variations of the algorithm have been created: one that involves just the cross-ratio, another which has a post-processing intensity check, and another that uses the intensity property as an integral part of the matching process. The first two are essentially the same, but one has the intensity based post-processing verification inserted into the algorithm. The third involves a redefinition of the algorithm. A diagram of the relevant steps for the first two algorithms is shown in Figure 3-19. The optional stage is shown in a dashed-line box. The algorithms for each stage are as developed previously.

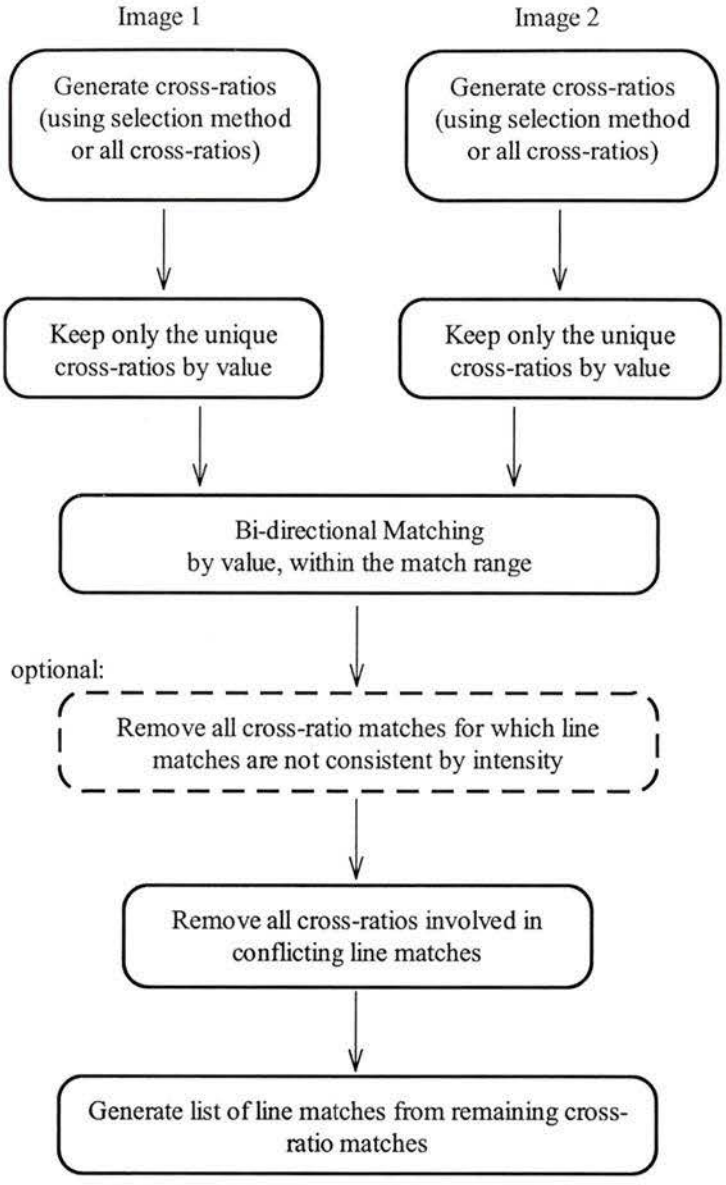

Figure 3-19: Complete Cross-ratio Based Matching Algorithm

Note that the preprocessing stages are identical for each vanishing point of each image and could be implemented as a stand-alone module. The flow diagram for the integral intensity property algorithm is shown in Figure 3-20.

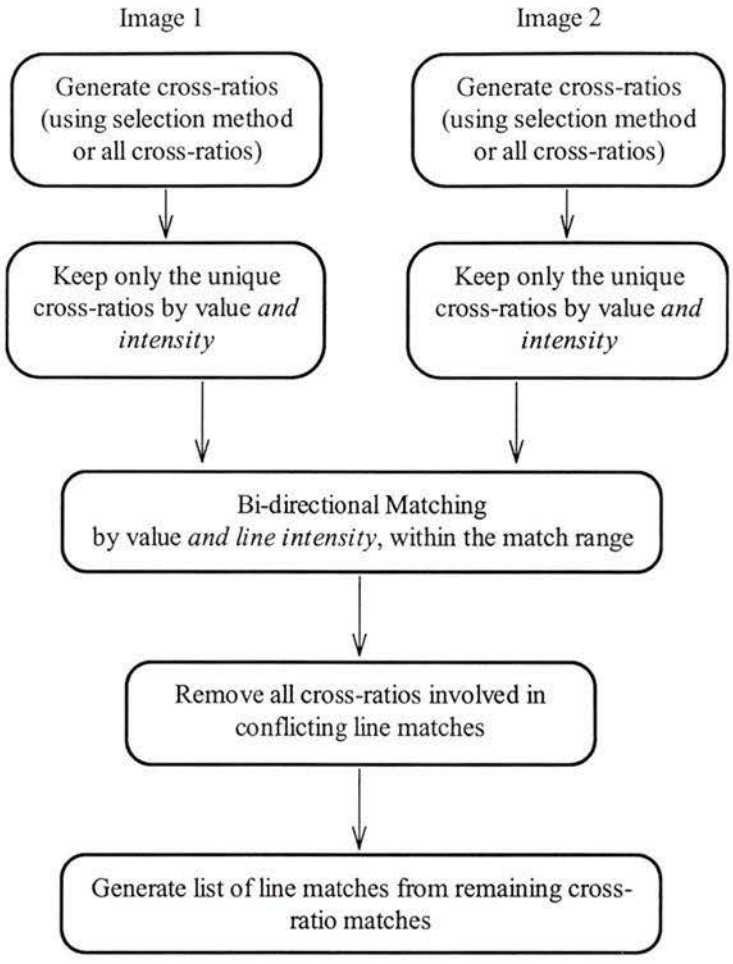

Figure 3-20: Integrated Matching Algorithm

The following parameters are used by the algorithms:

- T_M – Matching threshold – set equal to the maximum expected cross-ratio error
- T_U – Uniqueness threshold – set equal to the maximum expected cross-ratio error
- T_I – Intensity threshold – set loosely based on largest intensity property difference

The end-point alignment cross-ratio selection method also uses a parameter for the width of the end-point region.

The proposed algorithm was designed to match lines between two images, but can be extended to handle three images.

3.9 Conclusions

An algorithm for line matching has been developed. To satisfy the following step of camera placement, line matches must be obtained from at least 2 vanishing points, and restrictions exist on how the camera may move between frames, though they are not onerous. A line-matching algorithm was developed that included bi-directional matching, result consistency checking, a matching range requirement to only allow matches within expected noise levels, a uniqueness requirement to ensure clear matches, optional cross-ratio selection techniques to reduce data and increase the proportion of valid structures, optional intensity based match verification, and redefined uniqueness and matching requirements using the line border intensity.

4. Implementation Issues

The chapter focuses on the implementation issues that are not essential to the design of the matching algorithm. Issues of line orientation errors, segmented lines, and data overload are discussed and methods of dealing with them chosen. Actual test results will be described in the following chapter.

4.1 Line Orientation

There is error in the lines that are extracted. Radial lens distortion can rotate them, as can detection edge effects and imaging errors. The result is that the lines are generally slightly askew. In essence, none of the lines quite point at the intersection point, the vanishing point, which is a requirement for the cross-ratio. If the original line orientations are used, the resulting cross-ratios are occasionally below one or orders of magnitude larger than other cross-ratios in the image.

The stage of vanishing point detection essentially takes into account the errors inherent in the lines by considering all the lines [Magee84, Kittler93]. In essence, the individual errors in the lines are canceled in the process of determining a mutual intersection point. This information can be used to reorient the lines.

To correct line orientations, all lines were essentially rotated about their centroids to point directly at the calculated vanishing point location, thus ensuring the single intersection point. This was done simply by recalculating the equation of the line using the vanishing point and centroid as the two defining points of the line. Centroid errors occur, but they will tend to be along the length of the line which is perpendicular to the error being corrected. An example of this is shown in Figure 4-1.

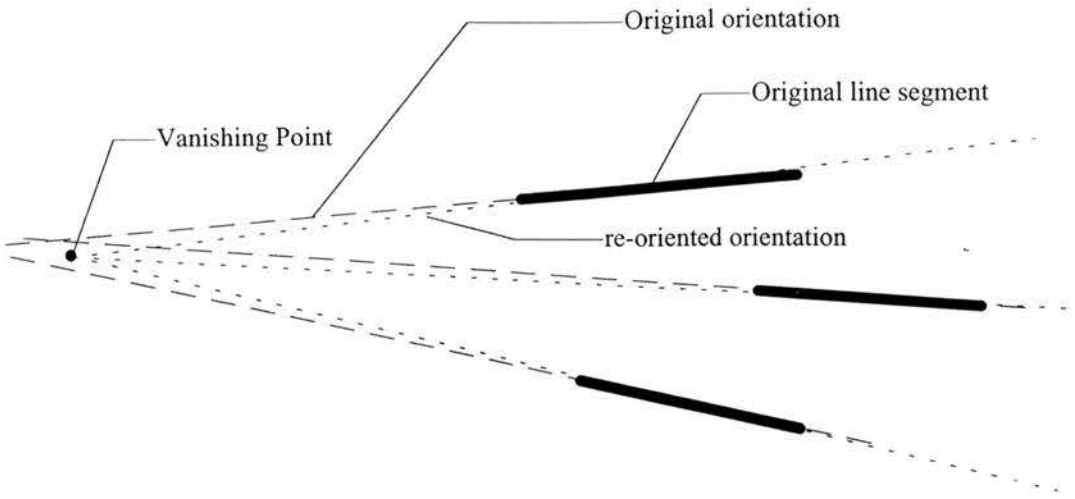

Figure 4-1: Line reorientation

4.2 Line Joining and Rationalizing

In real images, line intersections often break lines into separate segments. As well, some lines will be naturally aligned in 3-space. Calculating the cross-ratios for collinear line segments can result in a zero in the denominator (since the angle between lines is zero). These lines may also appear, due to noise as being very close instead of overlapping, and may generate very large cross-ratios that are essentially spurious. Both these cases must be dealt with when calculating the cross-ratios.

4.2.1 Line Joining

Line joining's goal is to re-join lines that were separated by the line finding process, usually at an intersection. They tend to be separated by a very small distance and the result is a single physical line where the two or more segments were previously. The characteristics of the line segments tend to be similar (absolute intensity on at least one side for example). Figure 4-2 is an example of a situation where line joining would be appropriate. Note that the end-points do not touch, and that errors in the line equation estimates may tilt them slightly, or even offset small segments.

Figure 4-2: Example of line segments for line joining

Line joining involves searching for lines that are roughly aligned and whose end-points are close but not overlapping, and then joining them, usually through their centroids. Various people have suggested methods of handling line joining [Liu91, Hussien93, Nacken93]. For our implementation, since we are only concerned with the lines associated with a vanishing point, we can join lines whose orientation varies by less than a threshold, and whose end-points are separated by less than another threshold. The orientation threshold is T_O and the separation threshold is T_S. The parameters will be determined by inspection of the data.

The ideal method of determining the new line is to refit a line to the line support region formed by the sum of the original line support regions. However, line support regions are no longer available, and joining centroids may result in a line that is not aligned with the vanishing point and is unacceptable. The new centroid position was defined as the line length weighted average of the original line centroids. The new centroid and vanishing point define the new parametric line equation. The new line length was defined as the sum of the original line lengths and the length of the gap separating them.

Line joining was implemented as a preprocessing step in the method. The definition of the algorithm is as follows.

$\forall i$

if $\exists j \ni (\angle L_i L_j < T_O)$ and ($|$ end-point separation $| < T_S$)

then join L_i and L_j

4.2.2 Line Rationalizing

Line rationalization involves joining lines that are simply aligned, not necessarily close or originally part of the same physical line. An example of this can be the top edges in a series of windows as is illustrated in Figure 4-3. They can be replaced by a single

parametric line. The line characteristics are not necessarily the same so only a parametric representation of the line is sensible. As well, the definitions of position and length can at best be applied to the shortest line segment the covers the original line segments. There is no physical meaning for the line.

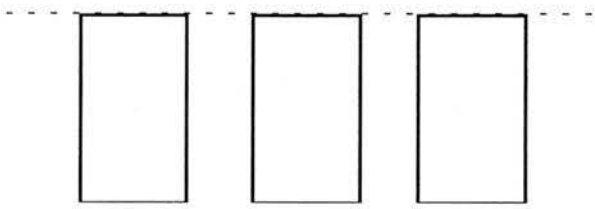

Figure 4-3: Example of line rationalization

Line rationalization is barely discussed in the literature, possibly because little work has been done considering relative lines where aligned and closely aligned lines could be a problem. The exception is the Hough transform [Hough62] which by definition finds all the aligned pixels and forces separation of lines by the bin spacing.

There are a number of issues involved in line rationalization. For matching, there is the question whether lines that are aligned in an image are actually aligned in 3-space. Figure 4-4 shows an example where lines may be rationalized which are not aligned in 3-space. In general, it can be assumed that they are [Lowe85], but the case of an error should be recognized.

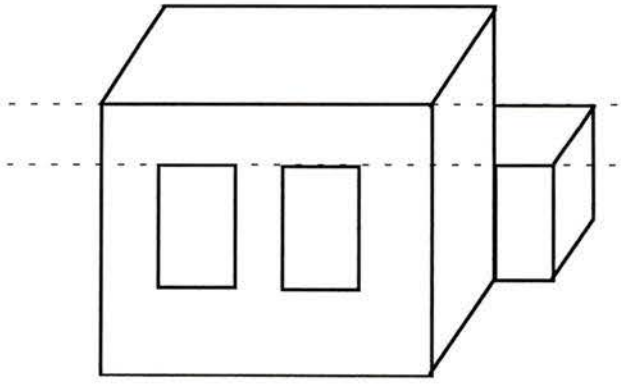

Figure 4-4: Example of mis-rationalized lines

Another important question is the threshold between aligned lines and separate lines. For example, in Figure 4-5, how many of the following segments get rationalized with segment A in either case. Usually, lines under consideration have slight angle and offset differences. In practice, the threshold is arbitrary.

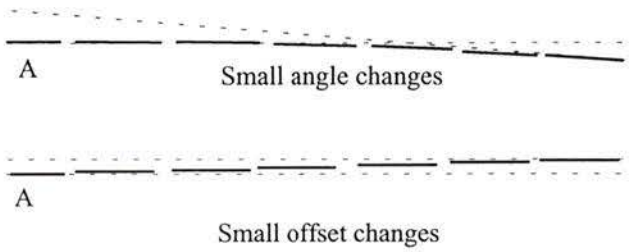

Figure 4-5: Line alignment threshold

Errors, such as radial lens distortion can result in lines not being rationalized that truly are aligned (misses), and the rationalization of others that are not (false positives). For this work, the only lines under consideration were oriented on the vanishing point so offset changes were not present. Lines could be rationalized strictly on the basis of angular difference. This threshold in effect determines the minimum angular difference between lines and hence the largest cross-ratio value that was possible. Since line rationalization was not intended to be an area of study unto itself, this threshold was chosen by inspection to join as many lines as possible that appeared to be aligned.

The process of rationalization – replacing two original lines by one lines – is also not obvious assuming the line equations are not identical due to error. Figure 4-6 shows two lines and one possible replacement line – the line joining the centroids. One of the lines could be arbitrarily used for the final line description (the longer one say), the line joining the centroids could be used, or some other weighted average of the lines could be generated. The ideal solution may be to refit a line to the line support region formed by the sum of the original line support regions.

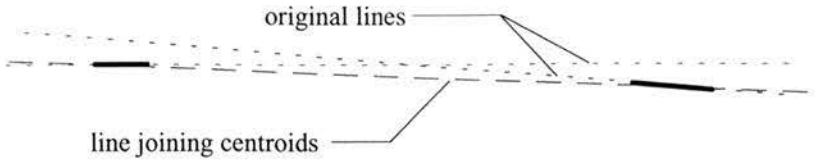

Figure 4-6: Rationalizing two lines with error

For this application, line support regions are no longer available, and joining centroids will result in a line that is not aligned with the vanishing point and is unacceptable. Three different options were used depending on other parameters of the program: using the longest line segment, keeping all segments and relabelling them with the same label, or using a length-weighted average of centroid positions joined with the vanishing point. Although all methods were implemented for testing, the final decision was simply to relabel all segments with the same label. Cross-ratios generated from the same lines were then filtered out after the entire list was generated. This was chosen so that segment positions would still be meaningful for cross-ratio selection methods.

4.3 Data Reduction Techniques

For images with hundreds of lines and the corresponding millions of cross-ratios, data reduction can be an implementation requirement. There are two general methods in which data is reduced: brute-force data reduction and uniqueness reduction. These two methods will now be discussed.

4.3.1 Brute force data reduction

There are many brute force methods of removing data. In general, it makes sense to keep the larger valued cross-ratios, since they tend to be less crowded and are less likely to get mismatched. Among the methods implemented are the following:

- maximum number of cross-ratios to keep
- fraction of cross-ratios to keep
- minimum cross-ratio value to keep

The maximum number of cross-ratios to keep essentially specifies an upper limit on the amount of data to keep. Unfortunately, the effective cut-off point in each image can be different, due to distribution changes and the total number of cross-ratios. For an image with a large number of cross-ratios, only higher valued ones will remain. In a matching image, this leaves a number of cross-ratios guaranteed without matches. These extra cross-ratios can be automatically removed during the matching process to further reduce processing.

The fraction of cross-ratios to keep is similar to the above, but is likely to result in ranges with more overlap, assuming cross-ratio ranges are similar and only the density of values changes. It can be used with an absolute minimum to keep for when the set is small to begin with.

The minimum value to keep is probably the most sensible measure to use. It ensures the same cut-off point in each image, so no images are left with unmatchable cross-ratios because their partners have been removed.

All three techniques were implemented as optional data filters, to be used when needed. These techniques are beneficial for immediate data reduction, even during the calculation of cross-ratios – removing them before the values are added to the list.

4.3.2 Uniqueness data reduction

The process of requiring "unique" cross-ratios by requiring no similar values within the uniqueness range as discussed in section 3.5.2 is inherently a form of data reduction. For example, consider the maximum number of cross-ratios, N, that can exist in the range 1.0 to R, if each is to be $U/100$ percent unique. The first invariant can be 1.0. The second invariant must by (1+U) times larger than this if it is not to be in the uniqueness range. The third invariant must be (1+U) times larger than that, and so on. The relationship is essentially the following:

$$(1+U)^N = R$$

Or more usefully as

$$N = \frac{\log(R)}{\log(1+U)}$$

When explored further, this limit itself effectively reduces the data as much as is required. If cross-ratios ranged in value from 1 to 20 (i.e. R=20), for a uniqueness range of 2%, there can be at most 151 values. Even for a 0.5% uniqueness requirement, only 322 cross-ratios can exist. This is more than sufficient data reduction. Unfortunately, uniqueness reduction happens after all cross-ratios have been calculated. The previous techniques may still be required for earlier reduction of large data sets.

Note that more cross-ratios in the original data set will actually result in fewer cross-ratios passing the uniqueness requirements. Twice the number of allowed cross-ratios, all equally spaced, would result in no cross-ratios passing the uniqueness requirement at all. The impact of this definition will be considered during the discussion of test results.

4.4 Edge effects

Data reduction techniques generally cut off the data at some value. There may an edge effect noticed with the end cross-ratios that are kept – they may appear unique when in fact their neighbours were simply removed by data reduction. To avoid this, the lowest cross-ratio is simply ignored for matching. This effect is illustrated in Figure 4-7.

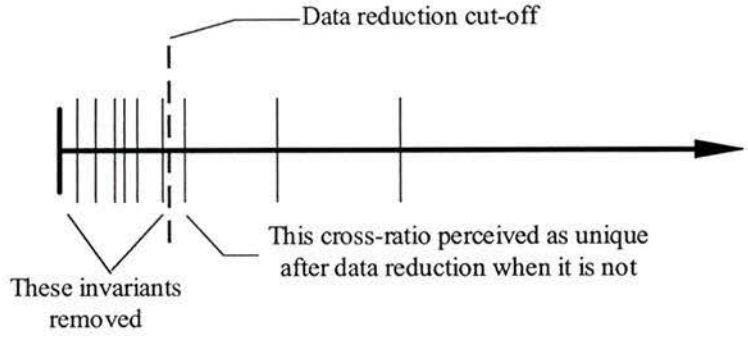

Figure 4-7: Edge effect

4.5 Summary

This chapter has presented implementation issues that were not immediately germane to the development of the matching algorithm. Lines were reoriented with the vanishing point to handle errors in line orientation, broken lines were rejoined and aligned lines were rationalized by relabelling the segments identically. Data reduction techniques were also presented, including the inherent data reduction of the uniqueness requirement. Lastly, edge effects were noted and dealt with by ignoring the lowest cross-ratio value for matching in each set.

5. Test Results and Analysis

This chapter describes and analyses the testing performed. The creation and purpose of each data sets is presented. The inherent error of the implementation and of the cross-ratio is determined, the effects of line joining and rationalization are shown, as is the amount of cross-ratio noise. The results of data selection, data reduction, and the uniqueness requirement are covered, followed by the actual match results.

5.1 Test Data

Tests were done to observe and analyze the algorithm in varied situations. Tests were designed to determine the algorithm's ability to handle complexity, cross-ratio "noise", measurement noise and camera movement.

Complexity refers to the number of objects and lines in a scene. Visually obstructing objects and repeated structures, which should generate identical cross-ratio values, correspond to complex data.

Cross-ratio "noise" refers to the inherent distribution of invariant cross-ratio structures and variant cross-ratio structures. This was observed by using image subjects that varied from uni-planar to multi-planar. Uni-planar refers to the existence of only one plane with enough lines for a cross-ratio structure for each vanishing point.

Measurement noise refers to noise of the input data for the algorithm. Perfect data can be constructed from synthetic data, and noisy data from real images. This also allows the recognition of how much noise was inherent in the algorithm implementation and the pre-processing steps of line and vanishing point finding.

Various camera rotations and translations were attempted. Most test data consisted of four images with incremental camera movement between images. A particular image pair could be chosen for a large or small amount of camera movement.

Testing was done so that the results of intermediate stages could be observed to see their effects, such as from line joining and rationalization, cross-ratio choosing methods and the uniqueness range. Of particular interest are the number of valid and invalid cross-ratios resulting from the choosing techniques and uniqueness range. Correct line match results were manually generated for most of the test data for automatic evaluation of the algorithm. As well, lists of coplanar lines were manually generated to assess the number of invalid and valid cross-ratio structures at various points in the algorithm.

The known line-matches and coplanar information was also used to determine the difference in valid cross-ratio structures between images. This allowed measurement of the inherent error of the cross-ratio computation. As well, tests were designed to demonstrate the implementation functioned as expected.

The test data can be divided into two broad categories: synthetic and real image data. It is now described in roughly ascending order of difficulty for the algorithm.

5.1.1 Synthetic Image Data Generation

Synthetic data directly generated the line equations and attributes from a 3-D line definition of the scene using a pin-hole camera model. The model for the camera and 3-D line segments were defined. Camera positions were then specified for each set of resulting image lines. The data was arranged to cover an area roughly equivalent to a real image as defined in the next section.

Values were defined for each line's parametric equation, length, region size (length × width), centroid position, and a width which was identical for all lines and arbitrarily set to 3 pixels. All other attributes (e.g. border intensities) were set to zero to represent that they were undefined. This allowed the data to be used to test the cross-ratio aspects of the algorithm, but not the line border intensity extensions. Since line length is not an attribute which is used in the algorithm, some lines cross other lines in the image. The pin-hole camera modeller did not perform hidden line removal.

Vanishing point location was identical for both synthetic and real data. Vanishing points were located by first manually identifying two lines associated with each vanishing point. The software would then identify all lines that could intersect the initial vanishing point when rotated about their centroids within a certain threshold amount (usually 0.01 radians). These lines were associated with the vanishing point. An optimization routine was then employed to locate the point which required the least length-weighted rotation of line segments for all segments previously identified to intersect the point. Vanishing points matches were then manually identified for all image pairs.

Both synthetic data sets were used to check the functioning of the algorithm implementation and show operation under ideal conditions. They also had no measurement noise and, having few lines and no obstructing objects, were not complex.

There are two synthetic data sets: SYNTHCUBE and SYNTHHOUSE. They are now described.

5.1.2 SYNTHCUBE test data

The SYNTHCUBE data set was of a cube with a door and two windows "drawn" flush on one face of the building. The images are shown in Figure 5-1 and Figure 5-2.

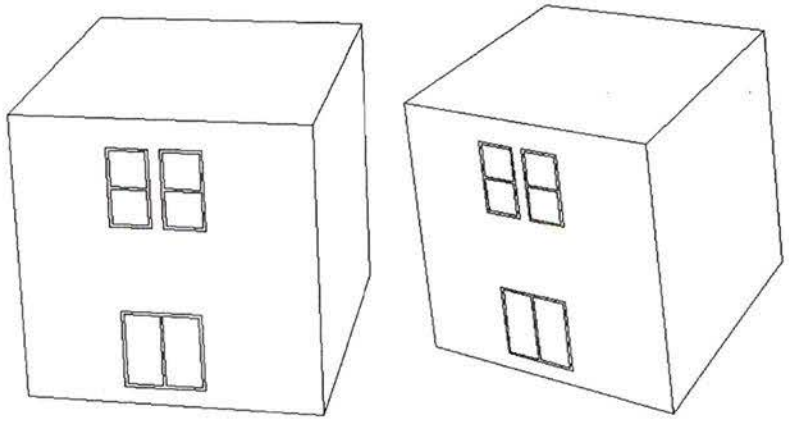

Figure 5-1: SYNTHCUBE 1 & 2 line images

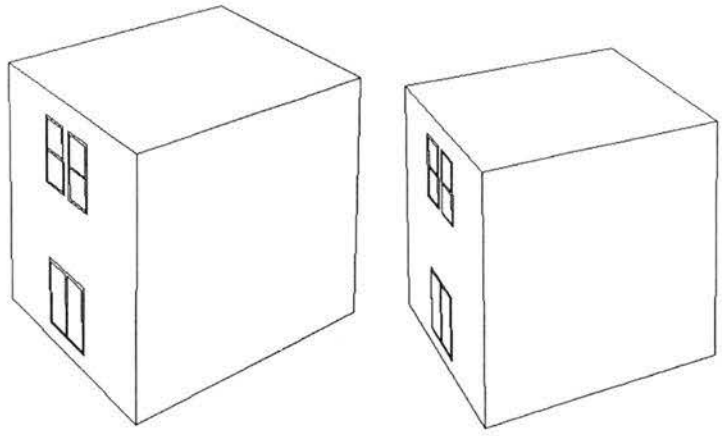

Figure 5-2: SYNTHCUBE 3 & 4 line images

The SYNTHCUBE data is uni-planar, so most cross-ratio calculations will be valid, invariant structures. It is designed to be similar to the LABCUBE real data set described later. It is the simplest data set developed.

5.1.3 SYNTHHOUSE test data

The SynthHouse data set is similar to the SynthCube, but it is multi-planar. Cross-ratios are no longer likely to be invariant. The windows are now recessed, as is the front door. The front door itself is on an projecting portion of the building. The line images are shown in Figure 5-3 and Figure 5-4.

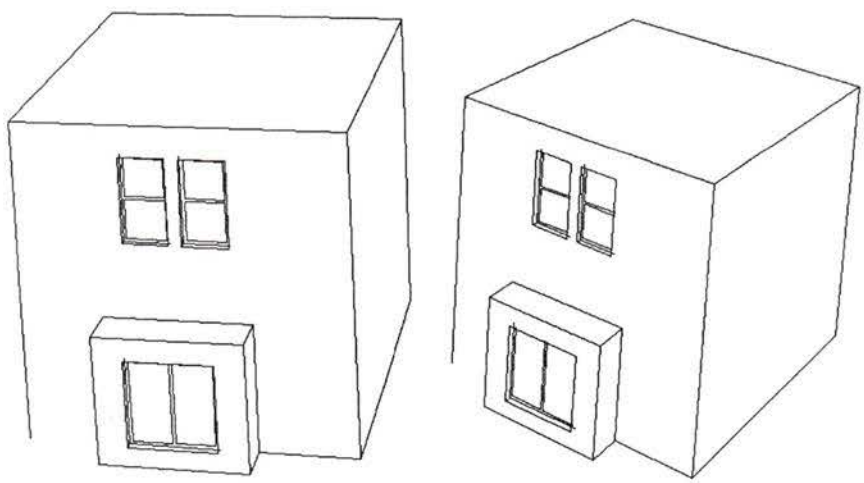

Figure 5-3: SYNTHHOUSE 1 & 2 line images

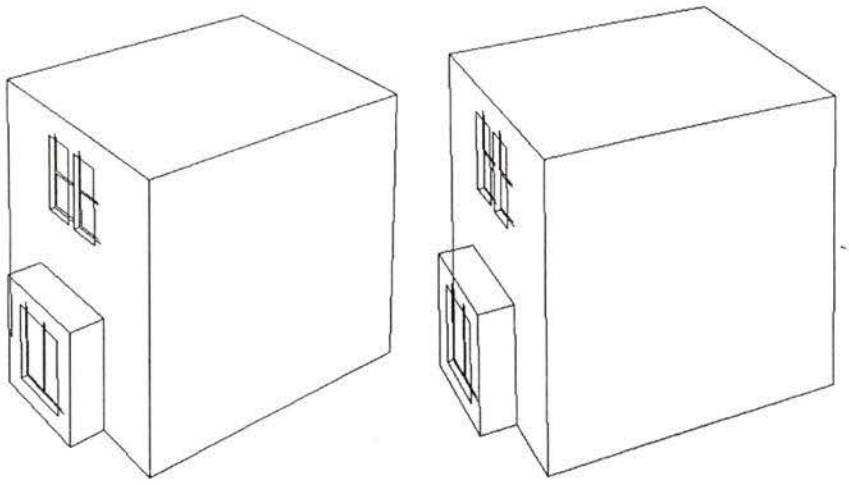

Figure 5-4: SynthHouse 3 & 4 line images

It will be noted that some lines appear to pass through other lines, and some line segments appear to be missing. This is of no consequence since only the parametric equation of the line is used in cross-ratio calculation. There is arguably some effect on some of the data choosing techniques, but their testing is really most appropriate on real images.

5.1.4 Real Image Data Generation

Real image data was obtained with a black-and-white Kodak DCS200 digital camera. The resulting images were 8-bit, 1024 by 1512 pixels and were reduced to be one half the size, 512 by 756 pixels, to facilitate processing. Lines were found with an implementation of the Burns line finding method described in chapter 2. Lines were filtered to keep only those that involve at least 30 pixels and a net intensity difference of 30. This reduced the amount of spurious lines (corresponding to textures like grass, and noise in the image). Vanishing points and matches were determined in the same way as for the synthetic image data described above. There were three real image data sets: LABCUBE, ELW and PETCH. They are now described.

5.1.5 LABCUBE test data

The LABCUBE test images were of a cube with windows and doors drawn on – a synthetic scene. They are similar to the SYNTHCUBE data set, allowing comparison between none

and typical levels of measurement noise. Cross-ratio "noise" is still low, though there are now other objects in the background. The images are shown in Figure 5-5 to Figure 5-8.

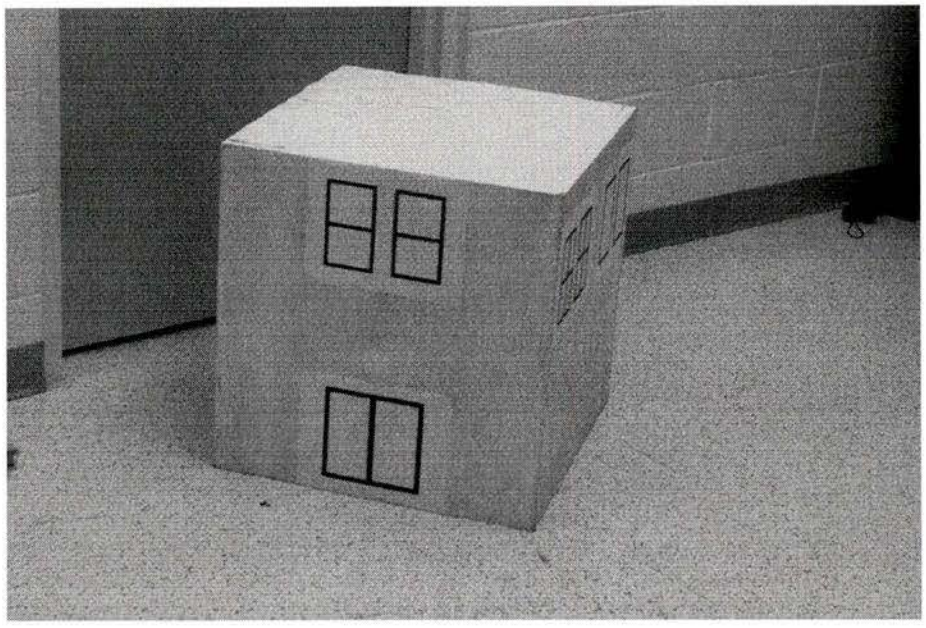

Figure 5-5: LABCUBE-1 image

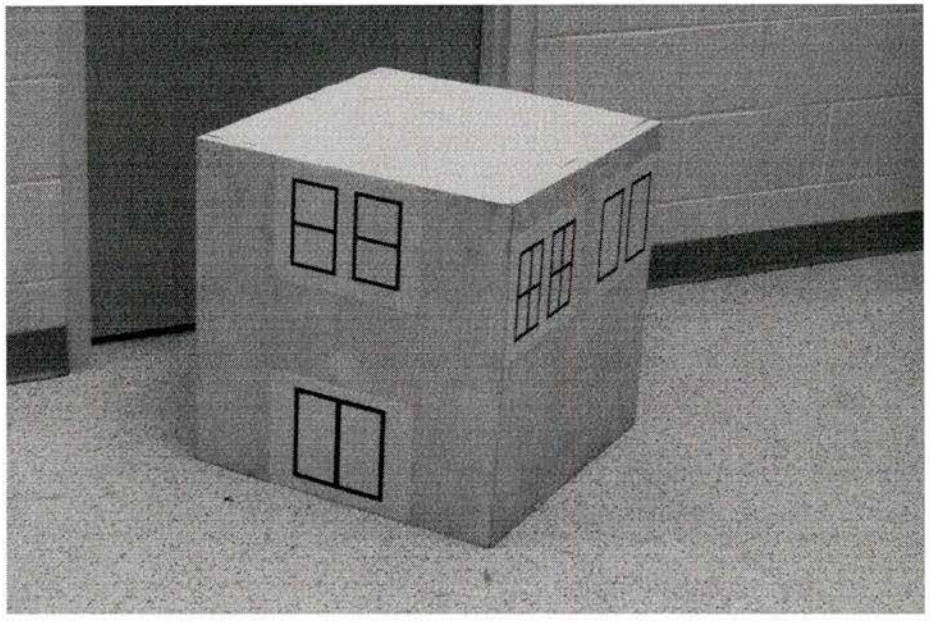

Figure 5-6: LABCUBE-2 image

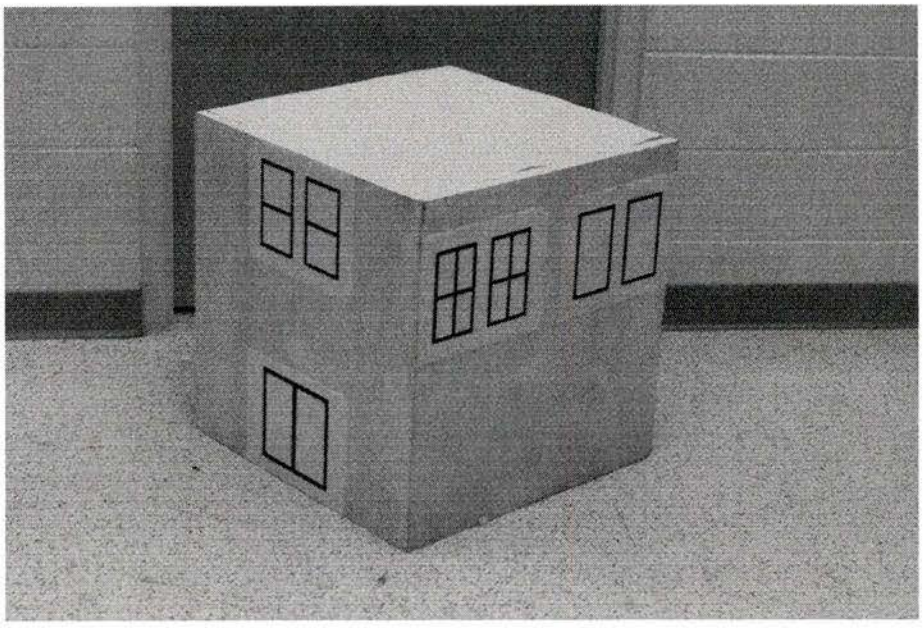

Figure 5-7: LABCUBE-3 image

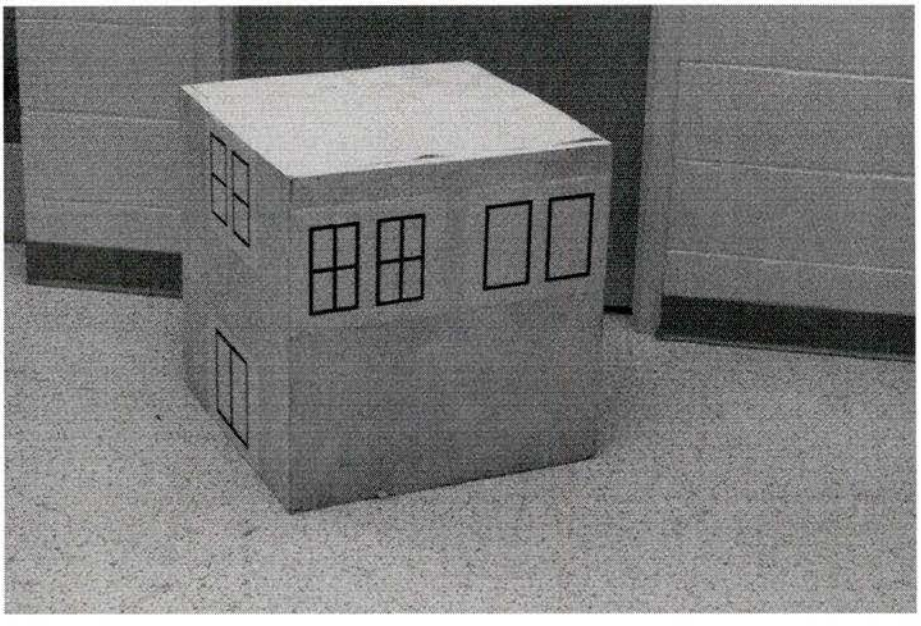

Figure 5-8: LABCUBE-4 image

And example of the line set for LABCUBE-2 is shown in Figure 5-9 and overlaid on the original image in Figure 5-10.

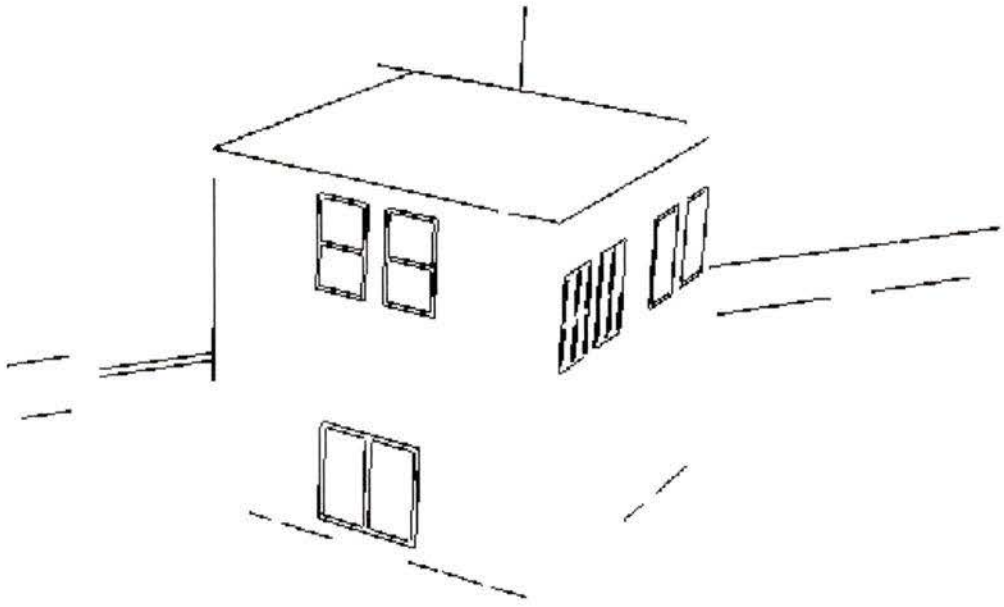

Figure 5-9: LABCUBE-2 line set

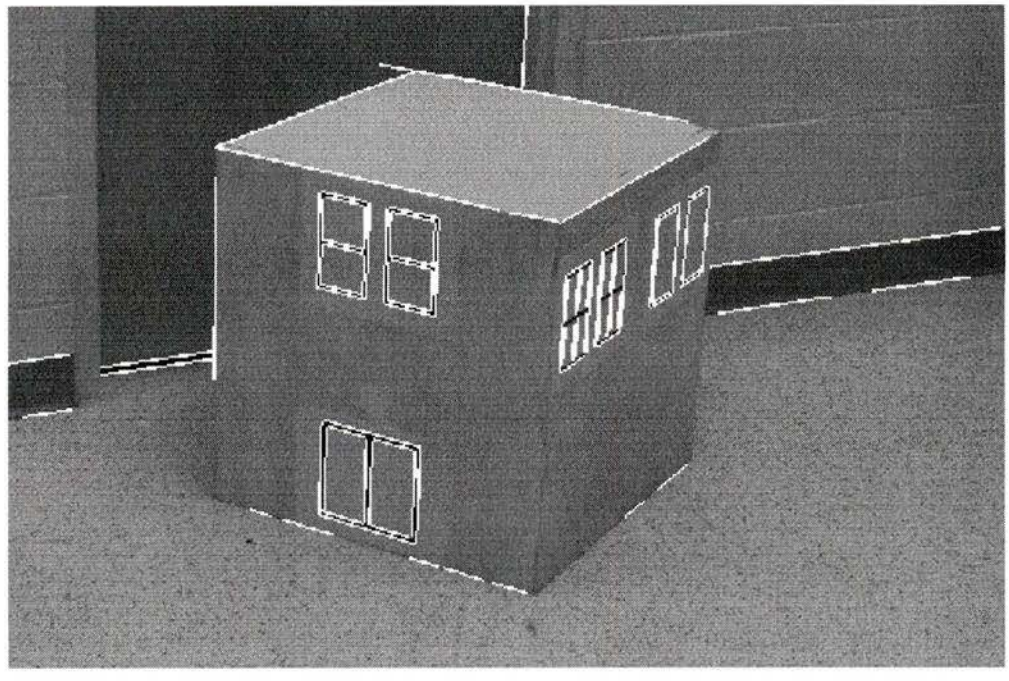

Figure 5-10: LABCUBE-2 image with overlaid line set

The LABCUBE data set shows aspects of real images that are not present in the synthetic data. In the real images, lines are occasionally separated, not necessarily at intersections, as is shown along the bottom edge of the cube (Figure 5-10). Some lines that would be obvious to an observer are missing in this image due to the thresholding of the lines. This was done to keep the data set simple and is a common practice. This also removed many short lines formed by the floor pattern. At the same time, the front corner of the cube has been completely lost because the intensity change across it was not large (less than 30).

The line corresponding to the back edge of the cube appear to extend past the cube. This is an error from the statistical method of calculating the centroid, implying that the line support region was not uniform in shape. This is not significant for cross-ratio calculations, though it may affect the use of that line by the cross-ratio choosing methods.

5.1.6 ELW and PETCH test data

These test images are typical of what the algorithm is supposed to handle. The ELW data set is a trio of pictures from the Engineering Lab Wing on the campus of the University of Victoria. This data set was chosen as a real and complex data set for the algorithm to handle. The data set is large and complex and was intended to be used only if tests on previous data went reasonably well. They also characterize the type of image that the algorithm must be able to handle. Correct line matches and lists of coplanar lines were not generated due to the large size of the data. Algorithm results would be checked manually. The images are shown in Figure 5-11 to Figure 5-13.

Figure 5-11: ELW-1 image

Figure 5-12: ELW-2 image

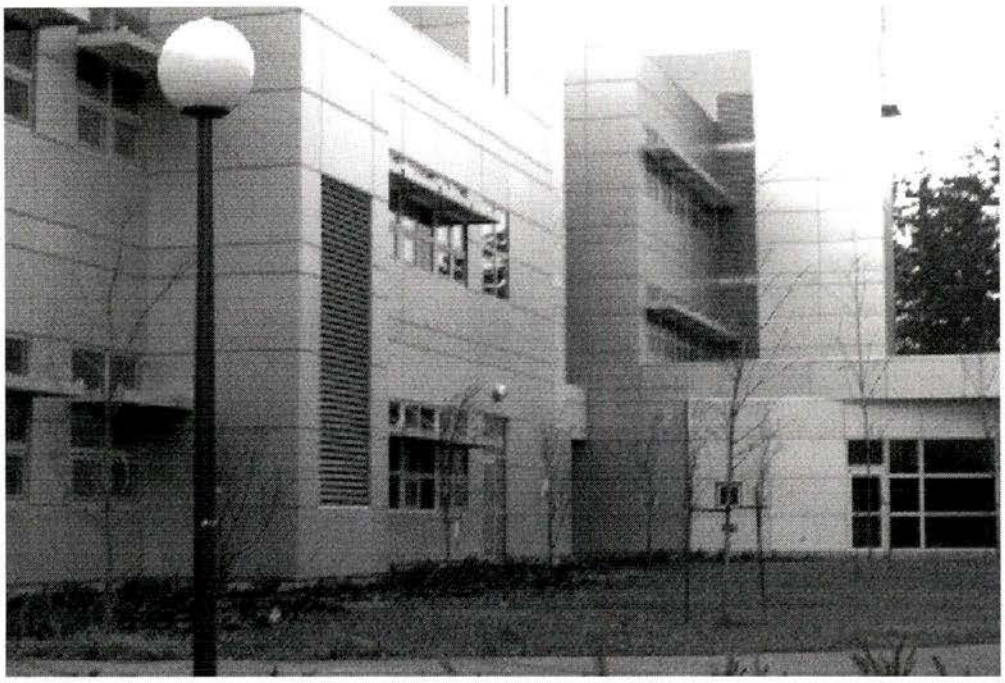

Figure 5-13: ELW-3 image

This data set is interesting in that most surfaces are flat with many lines on each plane, usable in invariant cross-ratio structures. An image of ELW-2 with overlaid lines is shown in Figure 5-14.

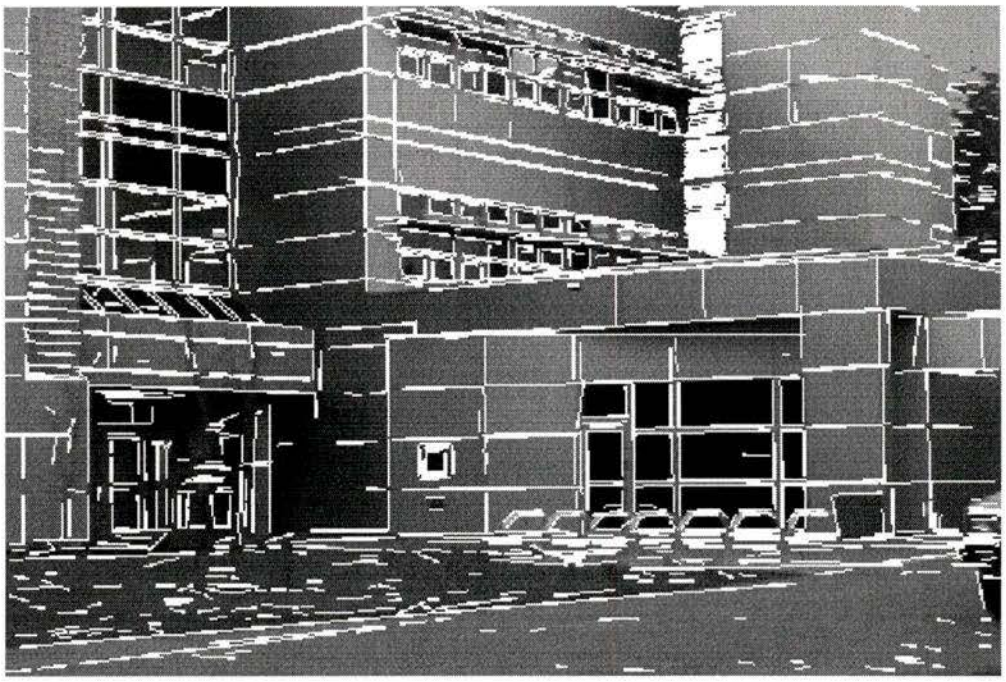

Figure 5-14: ELW-2 image with overlaid lines

Line centroid position errors are again visible in this image. It should also be noted that these images contain four vanishing points, the fourth from the tilted wall in the lower right of the image. Also of note is the large number of close lines in the upper right of the image that correspond to an air vent. Cross-ratios from this set of lines should span an entire range of values, possibly causing all of them to be considered non-unique.

The PETCH data set was chosen for the same reason as the ELW data set, except that it is characterized by relief texture. This data set consists of pictures of the Petch building at the University of Victoria. Images are shown in Figure 5-15 and Figure 5-16.

Figure 5-15: PETCH-1 image

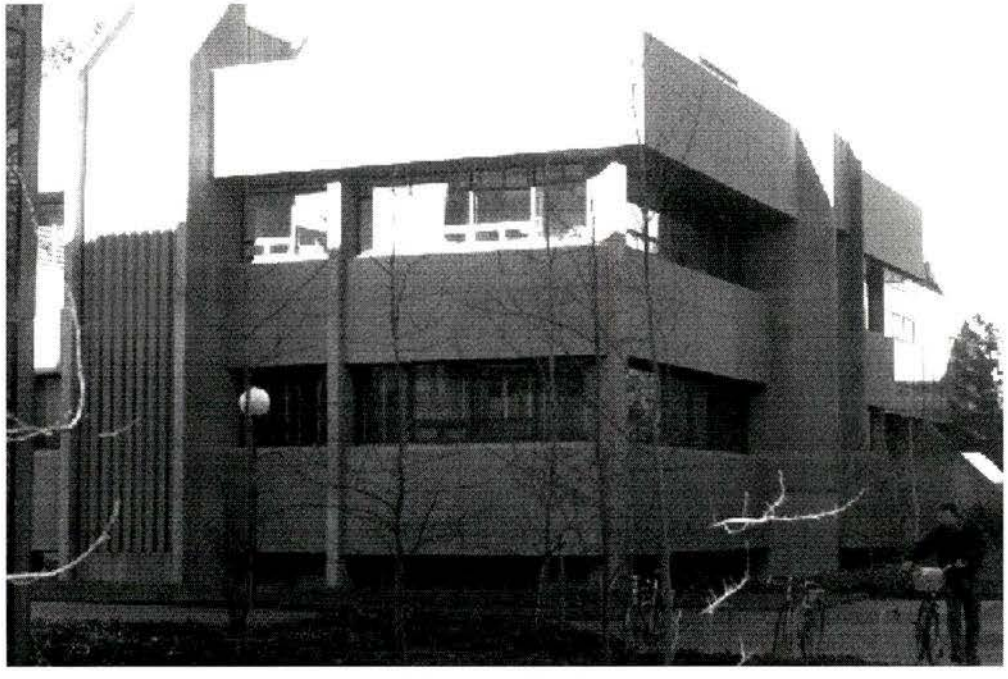

Figure 5-16: PETCH-2 image

Figure 5-17 shows PETCH-1 with overlaid lines. Of note here is that this image has only two identifiable vanishing points, though a third may be found in the slanted roofing. The partner image does not have enough lines to suggest a vanishing point for the root.

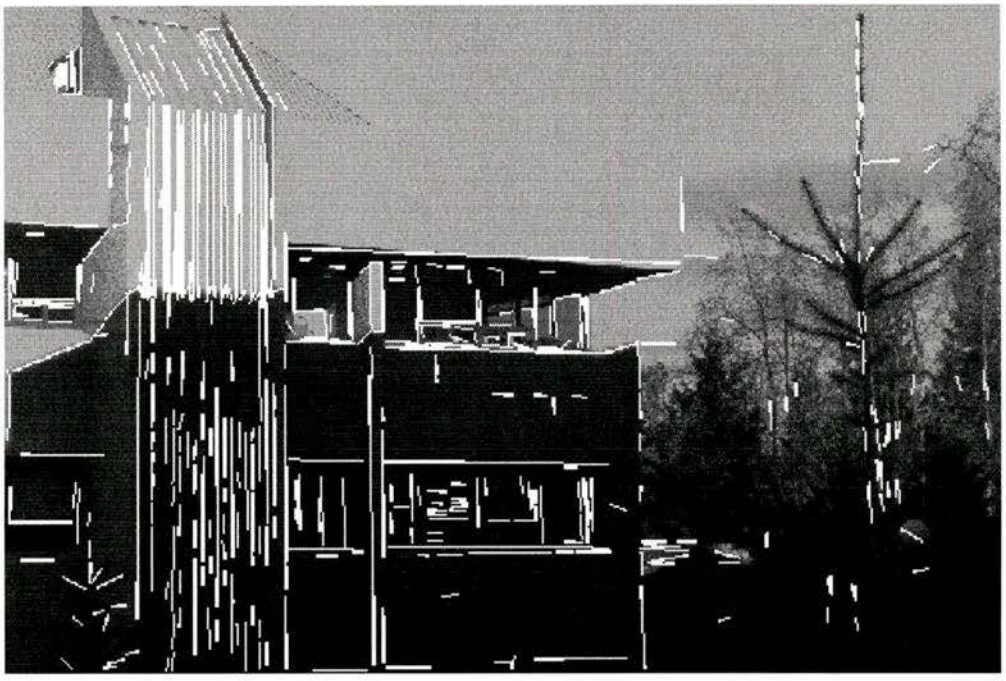

Figure 5-17: PETCH-1 image with lines overlaid

5.2 Cross-Ratio Analysis

A program was designed specifically to compare valid, matched cross-ratios between two images. The purpose was to explore the distribution and range of valid and invalid cross-ratio values, and error present in valid cross-ratio values.

The synthetic data sets, having been created directly from a definition of a scene and a model of the camera, were used to explore the inherent inaccuracies with the line description data format and calculation of cross-ratios. The SYNTHCUBE data set is sufficient for this purpose

The data from real images showed the level of error contributed by the imaging and feature extraction process. It also gives an indication of the range and distribution of cross-ratios in a real image. The LABCUBE data was used for this test.

The difference error between two cross-ratios, I_1 and I_2, in different images was measured as

$$Error = \frac{|I_1 - I_2|}{(I_1 + I_2)/2}$$

Error is measured relative to the cross-ratio value. Cross-ratios for valid cross-ratio matches in each image were calculated and the difference errors were tabulated.

5.2.1 Synthetic data tests

The summary of results for the SynthCube data set is shown in Table 5-1. The table shows each of two planes for which cross-ratios can be calculated for each image pair. The number of valid-cross ratios that can be calculated is either 1820 or 495. Cross-ratios ranged in value from just above 1 up to 33. The mean difference error for the synthetic data averages around 0.18% difference, never going higher than 1.41%.

Synthetic House Pair	Number of Lines	Number of Cross-ratios	Minimum Cross-ratio Value	Maximum Cross-ratio Value	Mean Difference (%)	Standard Deviation (%)	Maximum Difference (%)
1 & 2	16	1820	1.001043	25.523800	0.0490	0.0720	0.29
	12	495	1.000145	23.645677	0.0347	0.0731	0.39
1 & 3	16	1820	1.001043	25.523800	0.0740	0.1324	0.77
	12	495	1.000145	23.645677	0.0572	0.0854	0.50
1 & 4	16	1820	1.001043	25.523800	0.1794	0.2961	1.31
	12	495	1.000145	23.645677	0.0588	0.0837	0.48
2 & 3	16	1820	1.001039	25.459329	0.1742	0.2821	1.41
	12	495	1.000145	23.653869	0.0290	0.0276	0.15
2 & 4	16	1820	1.001039	25.459329	0.1713	0.2755	1.27
	12	495	1.000145	23.653869	0.0330	0.0326	0.18
3 & 4	16	1820	1.001048	33.515768	0.0637	0.1146	0.64
	12	495	1.000144	23.619385	0.0057	0.0087	0.05

Table 5-1: Cross-ratio Difference for SYNTHCUBE Data

This error is from the representation and round-off present in the pin-hole camera modeling program and the test program. This validates the cross-ratio as an invariant as implemented. It also shows that there is very little error from the implementation itself.

It was also noted that if the individual errors are plotted on a graph, they appear in definite quantized levels. Figure 5-18 shows an example of this. Each cross-ratio pair is plotted as a point along the horizontal axis by average value. The cross-ratio difference is plotted inverted as a percentage (the upper position was originally used to plot other information). The two plots correspond to the two planes in the data where cross-ratios could be calculated.

This quantization of the error suggests that it is some form of round-off.

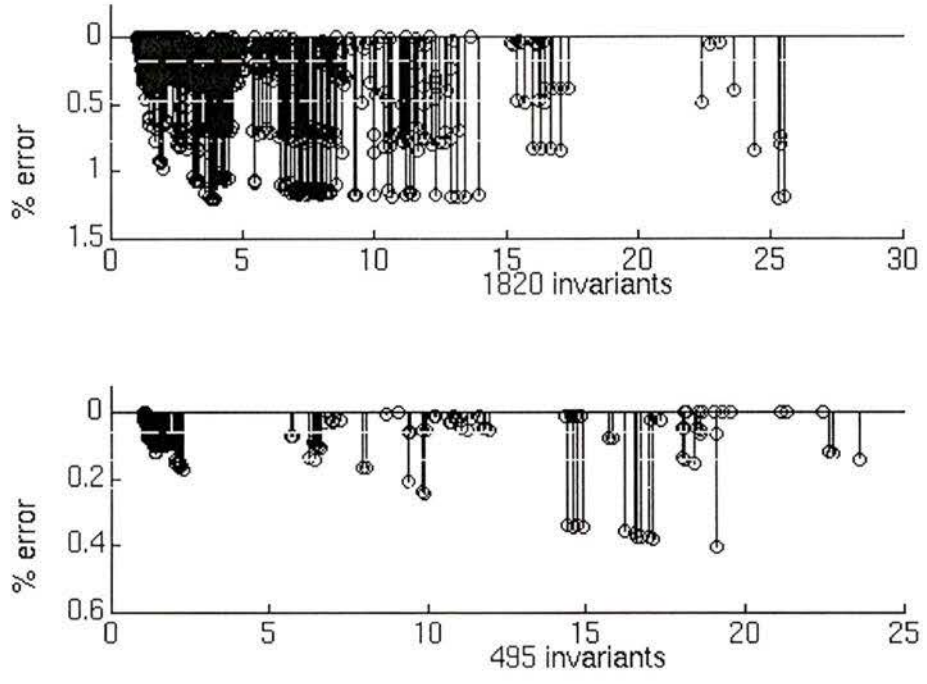

Figure 5-18: Difference error for SYNTHCUBE images 1 and 4 (by plane)

5.2.2 Real Data tests

The real image test data gives an idea as to the range and distribution of real cross-ratios. An example distribution of cross-ratios is shown in Figure 5-19. Cross-ratio differences are spread randomly and do not show the quantization levels of the synthetic data. This is the effect of the added measurement noise. Cross-ratio values tend to cluster at the lower end of the scale around 1, becoming more sparse when approaching the higher values.

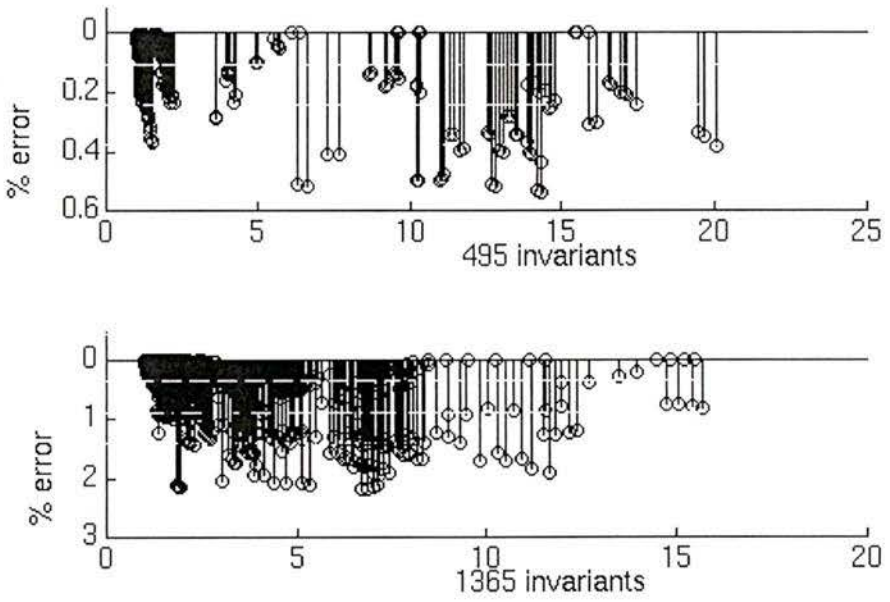

Figure 5-19: Cross-ratio Difference Error for LABCUBE images 1 & 2 (by plane)

The summary of results for the LabCube are summarized in Table 5-2. As expected for real images, the results are not as consistent as for synthetic data. Cross-ratio values spread from just above 1.0 up to 22.

In general, the mean error is low, around 1% difference. This is consistent with the findings of [Coelho92]. However, some image pairs show quite large errors: in particular, the image pairs 2&3 and 2&4 for the vertical lines of the right face and images 2&4 and 3&4 for the vertical lines of the left face.

House Image Pairs	Number of matched lines	Number of valid Cross-ratios	Minimum Cross-ratio Value	Maximum Cross-ratio Value	Mean Difference (%)	Standard Deviation
1 & 2	12	495	1.000197	20.079375	0.1049	0.1354
	15	1365	1.001034	15.705174	0.3453	0.5386
1 & 3	11	330	1.000197	19.669481	0.0735	0.1354
	15	1365	1.001034	15.705174	0.8707	1.7035
1 & 4	10	210	1.000256	19.669481	1.3692	2.7117
	8	70	1.004300	3.314523	3.3256	4.2961
2 & 3	11	330	1.000192	19.601361	0.1053	0.1532
	15	1365	1.001043	15.835122	0.7704	1.4503
	6	15	1.002089	21.652319	0.8671	1.3093
	17	2380	1.000731	14.743984	3.8653	7.7202
2 & 4	11	330	1.000192	19.601361	1.3427	2.5300
	8	70	1.004397	3.332419	3.4216	4.5151
	6	15	1.002089	21.652319	1.1369	1.4689
	17	2380	1.000731	14.743984	4.1508	8.2083
3 & 4	11	330	1.000191	19.556562	1.2540	2.4292
	8	70	1.004329	3.425939	4.0552	5.1844
	8	70	1.002008	22.403329	0.6747	0.6006
	20	4645	1.000162	19.678933	0.3068	0.7408

Table 5-2: Cross-ratio Difference for LABCUBE Image Pairs (by plane)

The first pairing suggests the problem lies with house 2 since it is common in both comparisons. This is in fact the first image where the side windows become visible. A close-up of the side-windows is shown in Figure 5-20. Lines that were not associated with vanishing points due to error are shown as dashed lines.

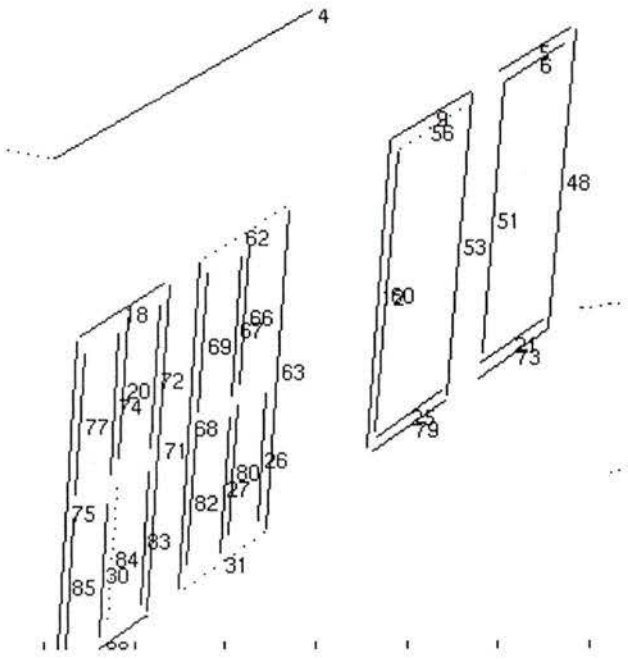

Figure 5-20: Close-up of LABCUBE image

Many of the vertical lines are either not present or were not associated with a vanishing point. Similarly, many of the horizontal lines are missing or were not associated with the vanishing point. This occurs because the cube face is nearly perpendicular to the image plane (see Figure 5-21). This results in the worst viewing angle for the resolving of features as their width decreases from the viewing angle.

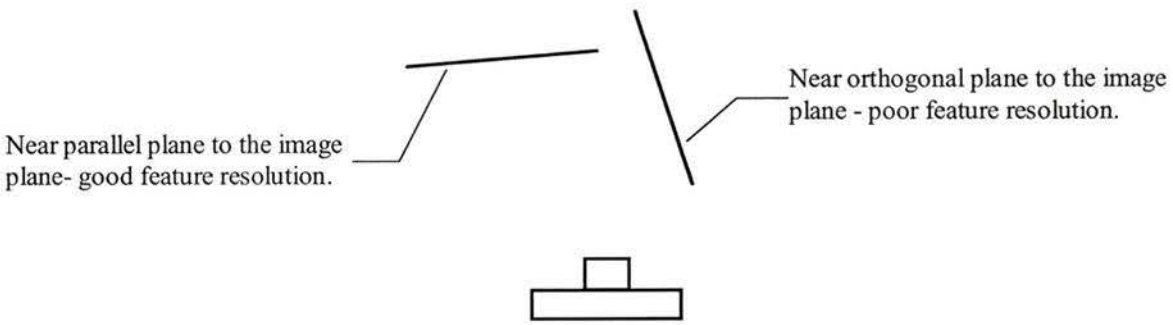

Figure 5-21: Illustration of parallel and orthogonal planes to the image plane

The second pairing suggests that house 4 is at fault. Again this plane of the cube is at its closest to being perpendicular to the image plane, suggesting that it is the same problem.

This suggests that errors are larger the closer that the plane is to being orthogonal to the image plane. This is sensible. Predicting this is more difficult. Basically, lines associated with a vanishing point close to the camera normal, and lines in the same region (assumed to be on the same plane) will have the highest error.

5.3 Line Border Intensity Analysis

The line border intensity difference could only be tested on the real data sets since the synthetic data did not have the intensities set. The LABCUBE data set was used to check the invariant aspect of this attribute. The results are show in Table 5-3. Results are shown for each plane of each pair that had enough lines to calculation cross-ratios. The intensity range of the line data is given, followed by the statistics for the smaller and larger line border intensity differences. The smaller difference is the one that is to be used as an indication of line similarity.

House Pair	Number of Lines	Intensity Range		Smaller Intensity Difference				Larger Intensity Difference		
		Min	Max	Mean	Std Dev	Min	Max	Mean	Std Dev	Max
1 & 2	8	56	238	3.8	1.9	1.0	6.0	8.6	4.7	15.4
	15	66	182	4.7	2.9	0.9	9.4	10.7	6.5	25.1
1 & 3	7	61	248	3.5	2.5	0.4	7.4	8.5	2.6	12.7
	15	69	190	9.8	4.3	3.1	19.3	24.7	13.4	45.5
1 & 4	6	71	244	1.7	1.1	0.4	3.3	11.3	7.6	23.5
	8	69	186	10.2	5.6	3.7	19.8	55.5	22.0	91.9
2 & 3	11	56	248	1.0	0.6	0.0	1.9	6.8	5.6	21.8
	15	66	190	4.2	3.0	0.1	11.3	21.2	12.9	38.3
	6	72	244	1.2	1.5	0.0	3.9	11.0	4.6	15.1
	12	78	190	2.7	3.6	0.1	11.8	27.9	5.1	36.2
2 & 4	11	56	244	2.3	1.4	0.1	4.3	16.0	11.4	40.7
	8	66	186	3.2	2.9	0.6	9.5	60.5	31.5	99.7
	6	67	238	2.5	2.6	0.1	6.5	15.9	15.2	37.6
	17	64	190	7.5	2.2	4.8	13.9	40.3	5.3	50.8
3 & 4	11	61	248	3.2	1.3	1.5	5.1	18.3	10.3	30.1
	7	80	190	2.0	1.1	0.4	3.5	36.2	16.6	64.4
	8	67	244	4.3	2.1	1.7	7.5	13.0	9.4	26.5
	20	64	188	5.9	3.2	0.2	13.2	19.5	8.4	34.3

Table 5-3: Line Border Intensities Differences for LABCUBE images

The results endorse the smaller intensity difference as a measure of line difference. The average smaller difference is approximately three intensity levels, whereas the average larger difference closer to 20, but ranges from 8.5 to 60.5. Some individual values are particularly large, even for the smaller differences. This suggests that the surface is not perfectly Lambertian. An intensity difference of 10 levels would keep almost all valid line matches and remove many of the invalid matches. This value will be used in the matching algorithm.

5.4 Preprocessing

This section looks at the preprocessing stages of the algorithm which occur before the calculation of cross-ratios. Of primary interest is the ratio of valid to invalid cross-ratio structures that are present in the data. Of secondary interest is the amount of line joining and rationalization that occurs.

Table 5-4 summarizes the preprocessing stages for the SYNTHCUBE data set. Each row corresponds to a vanishing point of each image. The second column lists the number of lines associated with the vanishing point. Columns three and four show the number of lines joined and rationalized, followed by how many unique lines remain after these steps. The next columns show how many cross-ratio structures can be formed from the remaining lines and how many are valid. The last column gives the signal to noise ratio (SNR) of the valid to invalid cross-ratio structures. Some entries are blank, signifying that there were insufficient lines to calculate a cross-ratio.

Image	Lines				Cross-ratios		
	Total	Joined	Ration -alized	Left	Total	Valid	SNR
1	21	2	6	13	715	495	2.25
	3	0	0	3	0		
	21	4	0	17	2380	1820	3.25
2	21	2	6	13	715	495	2.25
	3	0	0	3	0		
	21	4	0	17	2380	1820	3.25
3	21	4	4	13	715	495	2.25
	3	0	0	3	0		
	21	4	0	17	2380	1820	3.25
4	21	8	0	13	715	330	0.86
	3	0	0	3	0		
	21	4	7	10	210	126	1.5

lines rationalized/joined if angle difference < 1.00e-04 radians
lines joined if separation < 10 pixels

Table 5-4: Preprocessing for SYNTHCUBE House

The "Lines" section of the table shows that line joining and rationalization is occurring. These correspond to the window panes and doors that are side by side and aligned. It should be noted that this net reduction in the number of lines causes a large reduction in the number of cross-ratios that must be calculated. The "Cross-ratios" section of the table suggests that there are many valid cross-ratios in the resulting data. This is to be expected since the subject is uni-planar and few lines do not lie in a single plane.

Table 5-5 shows the same data for the SYNTHHOUSE data set.

Image	Lines				Cross-ratios		
	Total	Joined	Ration -alized	Left	Total	Vali d	SNR
1	27	4	0	23	8855	56	0.01
	27	2	10	15	1365	5	0.00
	9	0	1	8	70	0	0.00
2	22	2	1	19	3876	37	0.01
	23	1	6	16	1820	7	0.00
	9	0	0	9	126	0	0.00
3	22	2	2	18	3060	37	0.01
	23	4	3	16	1820	7	0.00
	9	0	0	9	126	0	0.00
4	22	2	5	15	1365	16	0.01
	23	7	0	16	1820	6	0.00
	9	0	0	9	126	0	0.00

lines rationalized/joined if angle difference < 1.00e-04 radians
lines joined if separation < 10 pixels

Table 5-5: Preprocessing for SYNTHHOUSE

This data set too has lines being actively joined and rationalized. Of particular importance is the cross-ratio section, especially the signal-to-noise ratio (SNR). This data set is multi-planar which has a large effect on the number of valid cross-ratio structures that can be found. The SNR is now never higher than 0.01. This implies that the algorithm will have a very difficult time finding valid cross-ratio structures and their match. This is much worse than anticipated.

Table 5-6 shows the results of the preprocessing for the LABCUBE real image data. This data set is more uni-planar than the SYNTHHOUSE, but less so than SYNTHCUBE.

Image	Lines				Cross-ratios		
	Total	Joined	Ration -alized	Left	Total	Valid	SNR
1	3	0	0	3	0		
	16	0	6	10	210	15	0.08
	21	4	0	17	2380	1365	1.34
	3	0	0	3	0		
2	4	0	0	4	1	0	0.00
	43	10	0	33	40920	3745	0.10
	12	2	1	9	126	1	0.01
	25	4	7	14	1001	210	0.27
3	7	1	2	4	1	0	0.00
	49	12	0	37	66045	4930	0.08
	35	6	10	19	3876	5	0.00
	20	2	6	12	495	126	0.34
4	10	2	2	6	15	0	0.00
	44	10	1	33	40920	3946	0.11
	39	10	14	15	1365	5	0.00
	19	3	3	13	715	126	0.21
				Total:	158070	14474	0.10

lines rationalized/joined if angle difference < 1.00e-04 radians
lines joined if separation < 10 pixels

Table 5-6: Preprocessing for LABCUBE data

Once again, line joining and rationalizing are actively occurring. The SNR of the data averages 0.1, and it is below 0.35 with one exception. The SNR will be correlated with the match results later in this chapter.

5.5 Cross-Ratio Selection Techniques

This section looks at the effectiveness of the cross-ratio selection techniques. There are four aspects upon which they could be judged: signal to noise ratio, number of valid cross-ratios retained, data reduction and matching results. The results of matching is clearly the real measure of success for the methods, but they will be discussed later with the rest of the matching results. The first three measures are now considered.

Table 5-7 shows a summary of the signal-to-noise ratio (SNR) for each of the cross-ratio selection methods for the SYNTHCUBE, SYNTHHOUSE and LABCUBE data sets.

Cross-ratio Selection Method	SNR		
	SYNTHCUBE	SYNTHHOUSE	LABCUBE
All	2.63	0.007	0.100
End-point alignment	4.85	0.011	0.447
Equi-radial Distance	2.39	0.014	0.264
Centroid-Line	2.44	0.013	0.325
Equi-radial & end-point	1.68	0.014	0.747

Table 5-7: Cross-ratio SNR for Various Selection Methods

It is clear that each of the methods has improved the signal-to-noise ratio, with the exception of the equi-radial and end-point alignment method for the SYNTHCUBE data. Which method is most effective is unclear since it varies for each data set. Nothing conclusive can be deduced except that the selection methods generally improve the SNR.

The percentage of valid cross-ratios retained by each method is given in Table 5-8. Obviously, using all the cross-ratios possible retains all the valid cross-ratios available.

Data Selection Method	Valid Cross-ratios Retained (%)		
	SYNTHCUBE	SYNTHHOUSE	LABCUBE
End-point alignment	21	42	77
Equi-radial Distance	19	45	71
Centroid-Line	31	52	76
Equi-radial & end-point	12	33	70

Table 5-8: Percentage of Valid Cross-ratios Retained for Various Selection Methods

The results suggest that the centroid-line technique is better than others, but not consistently or by a large amount. Generally, the percentage retained corresponds most closely with the choice of data set and this suggests that this measure is more indicative of the data itself than the selection technique. A clear trend is that the combined equi-radial

and end-point alignment selection technique retains the smallest percentage of valid cross-ratios, though not necessarily much less.

Table 5-9 shows the percentage of all possible cross-ratios that were calculated by each of the data choosing techniques.

Data Choosing Method	Percentage of Cross-ratios in Data		
	SYNTHCUBE	SYNTHHOUSE	LABCUBE
End-point alignment	27	30	24
Equi-radial Distance	28	26	33
Centroid Aligned	33	32	30
Equi-radial & end-point	16	18	16

Table 5-9: Data Reduction for Various Selection Methods

The amount of data reduction by the methods is broadly equivalent for the methods, although the equi-radial and end-point alignment methods clearly perform more data reduction than the others by a factor of two. In broad terms, Table 5-8 and Table 5-9 are similar, suggesting the proportion of valid cross-ratios retained corresponds to the amount of data reduction.

In general, no clear best method can be determined from the cross-ratio selection techniques. The equi-radial and end-point alignment method, since it is essentially the combination of two of the techniques, retains fewer cross-ratios and consequently, fewer valid ones.

5.6 Data reduction

This section discusses the effects of the brute force data reduction methods employed. The data reduction done by the cross-ratio selection methods has been discussed in the previous section. The reduction inherent in the uniqueness criterion will be discussed in the next section.

The only brute force data reduction technique employed was to remove all cross-ratios below a certain value. This is reasonable because so many cross-ratios are distributed near 1.0 that they will all be considered non-unique. As well, this treats all data sets equally, leaving them with the same data range. After some experimentation, a cutoff value of 2.0 was chosen. This had the effect of removing roughly 74% of the data without significantly changing the percentage of valid cross-ratios in the data. Table 5-10 shows how the LABCUBE data was affected.

	Initial Cross-ratios			Remaining Cross-ratios			
House Image	Number	Number Valid	% Valid	Number	Number Valid	% Valid	% Removed
1	0			0			
	210	15	7	76	4	5	64
	2380	1365	57	798	403	51	67
	0			0			
2	1	0	0	0			0
	40920	3745	9	10511	938	9	74
	126	1	1	55	0	0	57
	1001	210	21	226	43	19	78
3	1	0	0	0			100
	66045	5846	9	18208	1264	7	72
	3876	35	1	1116	10	1	71
	495	126	25	139	31	22	72
4	15	0	0	4	0	0	80
	40920	3946	10	9954	787	8	76
	1365	5	0	402	2	0	71
	715	126	18	234	22	9	67
All	158070	15420	9.76	41723	3504	8.40	73.6

Table 5-10: Effect of Cutoff on LABCUBE Data

The results for all the cross-ratio selection techniques and data sets are similar. Note that the percentage of valid cross-ratios does not vary substantially indicating that valid cross-ratios are not clustered at some range of values. This reduction of data allowed significant reduced data storage and calculation time.

5.7 Data Uniqueness Requirements

This section looks at the effect of the uniqueness requirement, based on cross-ratio value alone and also with the line border intensity based neighbour definition, on the data. Since

the LABCUBE data was shown to have up to 2% error in the cross-ratio matches, the uniqueness range (and match range) were set to 2%. This ensured that matches outside this range, likely not matches, would not be considered. Other ranges were tried and will be discussed shortly.

Table 5-11 shows the effects of the 2% uniqueness requirement (before and after) on the complete cross-ratio data for the LABCUBE data, which is indicative of the results for all the test data. Each row corresponds to a vanishing point in an image. The first column describes which image is under consideration. The next three columns describe how many cross-ratios there were and how many were valid, both by number and percent, before the uniqueness criterion was applied. The last three columns describe this same data again, after the criteria is applied.

House	Starting Cross-ratios			Unique Cross-ratios		
	Total	Total Valid	% Valid	Total	Total Valid	% Valid
1	0			0		
	76	4	5	14	2	14
	798	403	51	5	0	0
	0			0		
2	1	0	0	1	0	0
	10511	938	9	1	0	0
	55	0	0	20	0	0
	226	43	19	4	4	100
3	1	0	0	1	0	0
	18208	1264	7	3	0	0
	1116	10	1	4	0	0
	139	31	22	7	5	71
4	4	0	0	0		
	9954	787	8	0		
	402	2	0	15	0	0
	234	22	9	3	2	67

Table 5-11: 2% Uniqueness Requirement on LABCUBE Data

This table demonstrates the most important aspect of the algorithm. The uniqueness criteria decimates the data, sometimes completely. Only a handful of cross-ratios are left for each vanishing point. Whether they are valid or not (let alone have a match) appears fairly arbitrary. Note that the total number of unique cross-ratios seems to be inversely

related to the initial number of cross-ratios – the more initial cross-ratios, the fewer that are left. The effects of this will be more clear when the matching results are discussed.

Other ranges were tried, namely 0.25%, 0.5%, 1%, 3% and 5%. Their results are tabulated during the discussion of the matching results. A 0.5% uniqueness requirement implies roughly that 4 other unique cross-ratios could be within the error range of the calculation in the paired image. The general trend in the results is that larger uniqueness ranges result in a higher proportion of valid cross-ratios, but fewer of them. Differing ranges will be considered again in more detail when matching results are discussed.

This obliteration of data is largely because the cross-ratios are too crowded. It is for this reason that the use of intensity property was originally brought into the work. Now, cross-ratios are only crowded by neighbours whose lines are physically similar. Table 5-12 shows the effect of intensity based uniqueness on the cross-ratio data for the LABCUBE data.

House	Starting Cross-ratios			Unique Cross-ratios		
	Total	Total Valid	% Valid	Total	Total Valid	% Valid
1	0			0		
	76	4	5	22	4	18
	798	403	51	267	114	43
	0			0		
2	1	0	0	1	0	0
	10511	938	9	151	56	37
	55	0	0	26	0	0
	226	43	19	75	23	31
3	1	0	0	1	0	0
	18208	1264	7	339	0	0
	1116	10	1	47	1	2
	139	31	22	95	25	26
4	4	0	0	0		
	9954	787	8	56	18	32
	402	2	0	92	0	0
	234	22	9	55	20	36

Table 5-12: 2% Intensity & Value Based Uniqueness on LABCUBE Data

Using the definition of intensity neighbours has retained more cross-ratios. In essence, the intensity requirement has split the cross-ratio range into a series of different ranges,

depending on the intensities of each line. There is now more data on which to perform matching. Note that some vanishing points still lost all their valid cross-ratios. This tended to happen when there were very few to begin with or when the data was very dense. Considering intensity neighbours has only eased the problem of data decimation, not removed it. In essence, the less data there is before the uniqueness criteria is applied, the more will remain afterwards due to the reduced density. The data resulting from cross-ratio selection techniques is less dense, and therefore more likely to be unique.

5.8 Matching

The matching results are the most important test of the method. This section will cover the match results of the algorithm utilizing only the cross ratio (synthetic & real data) and including the line border intensity measure as a post-match check and as a part of the matching process (real data only).

5.8.1 Cross-ratio based matching

For all data sets, cross-ratios were removed for values less than 2.0. Tests were run with different cross-ratio selection techniques, match ranges and uniqueness ranges. The initial values of the match and uniqueness range to use were 2%, conservatively set by the observed error.

The first tests were performed on the SYNTHCUBE data set. This data set is uni-planar, so it was expected that most cross-ratios that could be formed would be valid. Table 5-15 shows the situation and matching results for the SYNTHCUBE data with a 2% matching and uniqueness ranges and no cross-ratio selection technique. Each pair of rows corresponds to two matched vanishing points, one from each image. The first column describes the image pair being matched. The next three columns describe the "unique" cross-ratios that are available for matching. The first of the columns is the total number of cross-ratios for each image, the second denotes how many matches exist in the combined list, and the third denotes what percentage of the data has a match present. The next two columns show the correct and total number of matches made (*correct/total*) in terms of cross-ratios and

actual line matches. The last section shows the number of matches after inconsistent line matches and their cross-ratios are removed. The last column gives the percentage of actual line matches that are correct. Blank entries correspond to insufficient initial data for the operation.

	Unique Cross-Ratios			Matches		Consistent Matches		
Image pair	Total	With matches	% correct	C-Rs	lines	C-Rs	lines	% lines correct
1 & 2	5 6	4	80 67	4/4	16/16	4/4	16/16	100
	4 8	0	0 0	0/0	0/0			
1 & 3	5 12	3	60 25	3/3	12/12	3/3	12/12	100
	4 1	0	0 0	0/0	0/0			
1 & 4	5 12	5	100 42	4/4	16/16	4/4	16/16	100
	4 12	0	0 0	0/0	0/0			
2 & 3	6 12	4	67 33	3/4	15/16	3/4	15/16	94
	8 1	0	0 0	0/0	0/0			
2 & 4	6 12	5	83 42	4/5	17/20	3/3	12/12	100
	8 12	0	0 0	0/0	0/0			
3 & 4	12 12	8	67 67	4/8	27/32	0/0	0/0	
	1 12	0	0 0	0/0	0/0			

Table 5-13: Match Results for SYNTHCUBE

This table is typical of the tests without any form of cross-ratio selection. Only two of the three vanishing points are shown for each image pair because there were no "unique" cross-ratios for the remaining vanishing point. For the second of the vanishing points shown, there are no valid matches to be made. No matches are found for this vanishing point. The first vanishing point has valid matches in the cross-ratio set. Some of these are

found for each image pair, resulting in at least three perfect cross-ratio matches for each image pair.

Image pair 2 and 3 demonstrates that one of the cross-ratio matches was incorrect, and from the data it can be deduced that it was only one incorrect line match in the cross-ratio match. The consistency check did not remove this pair – it is not fool-proof. A few incorrect matches would probably not harm the location of the camera positions, as long as there are sufficient matches that the relative noise is low. Some form of robust statistics may even identify and remove the bad match.

Image pair 3 and 4 demonstrate 4 of 8 bad cross-ratio matches. The consistency check removes these matches, but removes all the good ones as well, leaving no matches. This is typical of the action of the consistency check – a small number of bad matches are usually accepted, a large number cause much of the data to be removed.

Table 5-14 gives the match results for the SYNTHCUBE data, but with the centroid-line cross-ratio selection method used. This table has the same format as the previous one except that four columns have been added after the first column. These correspond to the calculated cross-ratios before the uniqueness criteria is applied. They are shown to give an indication of the cross-ratio noise in the original data, which is very little – as expected with uni-planar data. As before, only two vanishing points are considered since there are insufficient lines in the third to calculate a cross-ratio.

Calculated Cross-ratios				Unique Cross-Ratios			Matches		Consistent		
Total	Valid	Matches	% matched	Total	With matches	% correct	sets	pairs	sets	pairs	% lines correct
Image pair 1 & 2											
372	348	360	97	5	4	80	4/4	16/16	4/4	16/16	100
360	344		100	5		80					
504	380	380	75	12	3	25	3/4	15/16	3/4	15/16	94
622	491		61	13		23					
Image pair 1 & 3											
372	348	244	66	5	2	40	2/2	8/8	2/2	8/8	100
274	268		89	4		50					
504	380	304	60	12	1	8	1/1	4/4	1/1	4/4	100
759	654		40	10		10					
Image pair 1 & 4											
372	348	350	94	5	3	60	3/3	12/12	3/3	12/12	100
497	495		70	6		50					
504	380	15	3	12	0	0	0/0	0/0			
40	27		38	14		0					
Image pair 2 & 3											
360	344	243	68	5	2	40	2/2	8/8	2/2	8/8	100
274	268		89	4		50					
622	491	415	67	13	0	0	0/0	0/0			
759	654		55	10		0					
Image pair 2 & 4											
360	344	346	96	5	3	60	3/3	12/12	3/3	12/12	100
497	495		70	6		50					
622	491	17	3	13	0	0	0/0	0/0			
40	27		42	14		0					
Image pair 3 & 4											
274	268	270	99	4	1	25	1/1	4/4	1/1	4/4	100
497	495		54	6		17					
759	654	19	3	10	0	0	0/0	0/0			
40	27		48	14		0					

Table 5-14: SYNTHCUBE match results with centroid-line cross-ratio selection

This table is typical of all the cross-ratio selection techniques applied to this data set. Note that there are more cross-ratios for matching after the uniqueness criteria is applied. This is a direct result of the reduction of cross-ratios originally selected by the selection techniques. This translates into matches for the vanishing point that had no match results when all cross-ratios were used. One error still occurs, but there is no vanishing point with a large number of errors. This suggests that data reduction is important to determining matches.

The same test was performed with 1% and 3% uniqueness and match ranges. The 1% ranges resulted in even more unique cross-ratios and roughly double the number of correct matches made. Matches were found for all vanishing points but one. The 3% ranges resulted in fewer unique cross-ratios and matches and is quite similar to the original test with all cross-ratios. This indicates that reducing the data before the uniqueness criterion allows for less dense data, of which more is retained. Note that the error of this data set is particularly small and the ranges could be reduced much more for improved results.

The same tests were carried out on the SYNTHHOUSE data. This data is multi-planar, increasing the amount of cross-ratio "noise". It is reasonable to expect that the number of correct matches would decrease as the noise increased. This is indeed the case. An equivalent number of cross-ratios are involved in matching as for the SYNTHCUBE data. However, in the 2% data, there were usually two or no matches. When there were matches, they were usually inconsistent and almost no correct matches were determined by the method. The effects of smaller ranges corresponding to more cross-ratio data with more net matches in the data is the same However, almost no matches are made with other ranges, and those that were made were almost entirely incorrect.

Table 5-15 shows the matching results for all the cross-ratios with a 2% match and uniqueness ranges for the LABCUBE image data. This data summarizes the results from all the cross-ratio selection techniques. The first column shows the image pairs up for matching. Each row corresponds to the results for a different vanishing point. The columns show the results for each selection method. The first sub-column shows how many correct and total matches were made (*correct/total*). The second sub-column shows how many matches were present in the data and could have been made. Blank entries in the first sub-column signify that no matches were made. Blank entries in the second sub-column signify that there were not enough unique cross-ratios to perform matching.

Image pair	Vanishing Point	Cross-Ratio Selection Method									
		Centroid-line		End-Point Alignment		End-Point & Equi-radial		Equi-radial		All	
1 2	front		0	7/7	7	2/2	2	5/5	5	4/4	4
	up						0				0
1 3	front		0	2/2	5	2/2	2	4/4	6	4/4	5
	up		0		0				0		0
1 4	front	0/1	0		3		0		2		1
	up	0/1	0	0/1	0	0/1	0				
2 3	wall				0						0
	up										0
	side		0		0		0		0		0
	front	2/2	2	3/4	5	3/3	3	5/5	5	4/4	4
2 4	wall		0		0						
	up						0				
	side		0		0		0		0		0
	front		1		6	0/1	1		3	0/1	2
3 4	wall				0						
	up		0		0						
	side (R)		1		0	1/1	1		1		0
	front (L)	0/1	0		5	0/1	2		4		2

X/Y is X correct matches of Y attempted matches.
Next column shows number of matches present in the data.

Table 5-15: LABCUBE Matching results for all data, 2% match & uniqueness ranges

There are a number of things to note here. For each image pairing, there is only one vanishing point for which matches were present in the unique data, the horizontal vanishing point associated with the left face of the cube. Correct matches are found for most selection techniques except for the image pairs involving image 4. In image 4, the left face is most tilted relative to the camera plane and there is the most error in feature measurement. The image of the right face in image 2 is similar to that of the left face in image 4, suggesting that matches with it will also not be successful, which is indeed the case. However, with images 3 and 4, one match is found, and matches are available in the cross-ratio data, but less than for the left face. This could be due to the fact that the right face has only identical windows – fewer unique line combinations. A positive observation is that the matches that the algorithm generates are correct for the most part, though errors exist.

From this data set, the centroid-line cross-ratio selection technique appears inappropriate for this data set. The end-point alignment method has mixed success with the highest number of correct matches for an image pair, and many image pairs with no matches. The other two methods and simply using all cross-ratios seem equivalent.

The question arises whether more matches could have been made with smaller uniqueness and match ranges. This was tried and Table 5-16 shows the number of matches available for all the choosing techniques for various uniqueness ranges. Each row corresponds to a matched vanishing point. There are five major columns corresponding to the cross-ratio selection technique. Each of these is sub-divided into columns corresponding to different uniqueness ranges ranging from 0.25% to 3%. The actual entries show how many matches exist in the matching data. Blank entries exist where insufficient cross-ratios were available for matching. "Front" and "side" refer to the left and right horizontal vanishing points, respectively.

VP	centroid-line					end-point alignment					end-point & equi-radial					equi-radial distance					All cross-ratios				
	.25	.5	1	2	3	.25	.5	1	2	3	.25	.5	1	2	3	.25	.5	1	2	3	.25	.5	1	2	3
House Pair 1 & 2																									
front	0	0	0	0	0	10	9	7	7	5	2	2	2	2	2	7	6	5	5	4	8	7	5	4	4
up	0	0	0			1	0	0			2	0	0	0	0	0	0	0			0	0	0	0	0
House Pair 1 & 3																									
front	0	0	0	0	0	6	6	6	5	3	2	2	2	2	2	8	8	6	6	4	8	7	5	5	4
up	0	0	0	0		2	0	0	0	0	3	1	0			0	0	0	0	0	0	0	0	0	0
House Pair 1 & 4																									
front	0	0	0	0	0	4	3	3	3	3	0	0	0	0	0	2	2	2	2	2	3	3	3	1	1
up	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0		
House Pair 2 & 3																									
wall						0	0	0	0	0											0	0	0	0	0
up	0	0	0			2	0	0			11	1	0			0	0	0			0	0	0	0	0
side	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
front	27	18	6	2	2	32	20	10	5	3	6	5	3	3	3	13	11	6	5	4	42	26	9	4	4
House Pair 2 & 4																									
wall	0	0	0	0	0	0	0	0	0	0											0	0	0		
up	0	0	0			0	0	0			4	1	0	0	0	1	0	0			0	0	0		
side	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0
front	21	15	5	1	1	29	16	10	6	4	1	1	1	1	1	3	3	3	3	2	36	24	9	2	2
House Pair 3 & 4																									
wall						0	0	0	0	0											0	0	0		
up	1	0	0	0		1	0	0	0	0	3	0	0			0	0	0			0	0	0		
side	4	4	2	1	0	3	3	2	0	0	4	4	2	1	1	4	3	1	1	1	0	0	0	0	0
front	20	12	4	0	0	35	18	10	5	3	2	2	2	2	2	4	4	4	4	2	41	23	8	2	2

Sub-columns refer to different uniqueness ranges

Table 5-16: Effect of uniqueness ranges on available matches for LABCUBE

This table shows the relationship between uniqueness range and the number of potential matches that can be found. As the uniqueness range is reduced, the table shows an increase in the number of potential matches. This is due to an increase in the number of cross-ratios that are kept when the uniqueness range is decreased. Assuming that the ratio of valid and invalid cross-ratios do not change greatly with changes in uniqueness range, results like those shown in the table are not unexpected.

The corresponding matching results are shown in Table 5-17. The arrangement of the table is the same, except the match range is also listed, though it is identical to the uniqueness range in each case. The table entries now correspond to the number of correct

and total matches from the algorithm (*correct/total*). Entries are blank if no matches were made.

	centroid-line						end-point alignment						end-point & radial						equa-radial distance						basic - all invariants					
VP	.25	.5	1	2	3	5	.25	.5	1	2	3	5	.25	.5	1	2	3	5	.25	.5	1	2	3	5	.25	.5	1	2	3	5
House pair 1 & 2																														
front	0/1							8/8	7/7	7/7	5/5	1/1	0/1	1/1	2/2	2/2	2/2			1/1	5/5	5/5	4/4		2/2	7/7	5/5	4/4	4/4	1/1
up		0/1												0/1						0/1										
House pair 1 & 3																														
front	0/1						0/2	2/2	6/6	2/2	3/3	2/2		1/1	2/2	2/2	2/2		0/1	1/1	6/6	4/4	4/4			2/2	5/5	4/4	4/4	
up								0/1	0/1						0/1															
House pair 1 & 4																														
front	0/1			0/1		0/1	0/1	0/1			0/1										0/1		0/1							
up			0/1	0/1				0/1		0/1				0/1		0/1														
House pair 2 & 3																														
wall											0/1	0/1																		0/1
up								0/1							0/1					0/1							0/1			
side							0/1	0/1																						
front		3/3	2/2	2/2	2/2		6/6	7/7	4/4	3/4	3/3	1/1	4/4	5/5	1/1	3/3	3/3	1/1	2/2	7/7	6/6	5/5	4/4	1/1	4/4	4/4	2/2	4/4	4/4	
House pair 2 & 4																														
wall																														
up								0/1												0/1										
side							0/1																							
front	0/2	0/1	0/2		0/1			0/1			0/1					0/1	0/1				0/1		0/1				0/1	0/1	
House pair 3 & 4																														
wall																														
up																														
side	3/4	0/2			0/1		1/1		1/1						1/1	2/2	2/2	2/2	1/1				1/1	1/1				0/1		
front	0/1			0/1	0/1			0/1	0/1		0/1					0/1	0/1						0/1					0/1		

Match and Uniqueness ranges correspond to value at top of each sub-column

Table 5-17: Uniqueness & Matching range effects on matches for LABCUBE

The matching results do not show a significant difference in match results for ranges different from 2%. In fact, for the values farthest from 2%, there are almost no matches generated. Smaller ranges will allow more cross-ratios, but will make some valid matches no longer found because measurement noise has caused them to be separated by more than the range size. The uniqueness and match range of 2% is reasonable for the real images.

5.8.2 Cross-ratio and Intensity Based Matching

By adding the use of the line border intensity, the matching results should be able to be improved. There were two proposed methods of using this property: as a check of the matches, or as an integral part of the definition of uniqueness and identifying potential matches. Both these methods are now attempted. Only real data is used since the synthetic data was lacking intensity information. A matching and uniqueness range of 2% is used. An intensity range of 10 is used. This is loose and allows for surfaces to not be truly Lambertian. A tighter range may separate true matches. This is more likely to allow false matches than to lose matches.

Table 5-18 shows the match results for the LabCube data for all three combinations of properties: just the cross-ratio, with intensity match checking, and intensity and cross-ratio based matching. All matching was done with a 2% uniqueness and matching ranges and an intensity range of 10 where applicable. Again, each row corresponds to a vanishing point of an image pair. The major columns denote the different cross-ratio selection methods. Each major column is divided into three sub-columns. The first corresponds to the match results just using the cross-ratio, from the previous section. The second column is the same results checked by the line border intensity property. The third column uses the intensity property to determine uniqueness and which cross-ratios are considered for matching. Table entries give the number of correct and total matches made (*correct*/total). If no matches were made, the table entry is blank.

VP	centroid-line			end-point alignment			end-point & equi-radial			equi-radial distance			all cross-ratios		
House pair 1 & 2															
front			0/1	7/7	7/7	5/8	2/2	2/2	2/2	5/5	5/5	7/7	4/4	4/4	10/17
up			2/2			5/5			4/7			3/4			16/16
House pair 1 & 3															
front				2/2	1/1	2/2	2/2	2/2	2/2	4/4	6/6	8/8	4/4	5/5	8/13
up						0/1			0/1						
House pair 1 & 4															
front	0/1													0/1	
up	0/1			0/1			0/1								
House pair 2 & 3															
wall															
up						1/1						0/1			
side															
front	2/2	2/2	23/23	3/4	5/5	17/17	3/3	3/3	6/6	5/5	5/5	11/12	4/4	4/4	25/28
House pair 2 & 4															
wall															
up															
side															
front							0/1						0/1		1/1
House pair 3 & 4															
wall															
up															0/2
side		1/1	4/7			0/1	2/2	3/3	2/2						0/1
front	0/1						0/1								3/3

Table 5-18: Effects of intensity checking & uniqueness on LABCUBE data

Adding the intensity property to check matches made using only the cross ratio removes almost all bad matches resulting in near perfect matches. Interesting, by removing all matches where only one match was made for a vanishing point removes almost all errors as well. Making intensity an integral part of the matching process results in more matches, even where matches were not previously made, and some errors again. Using the intensity property to define uniqueness has effectively split the data, allowing more to be retained, including more correct matches. However, it is not enough to generally obtain matches in each image.

Comparing major columns, the centroid-line cross-ratio selection method seems least effective. Of the rest, the only one that stands out significantly is the combined end-point

and equi-radial method which generated matches for image pairs 3 and 4 for the right face of the cube. However, there is not enough evidence to make any firm conclusions on this.

The combined criteria of end-point alignment and equi-radial obtained matches where other techniques did not suggests that the increased data reduction of this method allowed more cross-ratios to be retained for the matching process. This corresponds to the problem of cross-ratio noise of invalid structures. This is the limiting aspect of the method. It has managed to determine matches for large camera movements, but not consistently produce an answer.

In general, when matches were made, they were correct. The problem is the blanks in the table where no matches of any kind were made. There are insufficient matches to determine the original camera positions in general. Since the PETCH and ELW data sets are multi-planar, whereas LABCUBE was uni-planar, results are expected to be worse. It was felt that there was little to be gained and these last two data sets were not run.

5.9 Conclusions

The algorithm did succeed in finding some line matches for a camera that rotated around an object for up to 70 degrees. No other method is known that can do this for unknown camera positions. However, there were usually insufficient matches for the goal of determining the camera positions. As well, it could not perform this consistently. The major reason for this was the amount of invalid structure noise in the list of generated cross-ratios – the signal-to-noise ratio became too high. If the proportion of valid cross-ratio structures could be increased, the method may be applicable. This could be accomplished by choosing another invariant structure that can be located reliably, or by one which occurred less often, then the data volume and structure noise would be low enough for the method to function. It is noted that the method tended to produce correct answers when matches were identified. Also, the algorithm does not appear overly sensitive to the specific value of parameters. Parameters are chosen in a sensible manor, based on basic tests of the setup. The signal-to-noise ratio was much higher than expected

and the results are in and of themselves surprisingly good considering. Although the method did not achieve the goal of allowing camera position determination, it did provide line matches for large, unknown camera movements with uncalibrated cameras.

The line border intensity measure was also shown to be effective in improving matching. As far as could be determined, this is the first time this property has been suggested or used in vision analysis. It was used to effectively either prove matches, or to separate the data into neighbours and non-neighbours, essentially reducing the density and making cross-ratios more unique.

6. Conclusions and Future Work

A method of determining line correspondences for unknown camera positions has been proposed in this thesis. The method makes use of the cross-ratio invariant from projective geometry and the minimum line border intensity difference of two lines. Neither the cross-ratio nor the minimum line border intensity difference is known to have been applied to this problem before.

The method indicated that line matches could be found for uncalibrated cameras in arbitrary positions (i.e. large motion). As well, it demonstrated the usefulness of the minimum line border intensity difference for matching. Experimental results showed that the signal-to-noise ratio was extremely low, on the order of 0.01 to 0.1 for real images. Although insufficient matches were found for the camera positions to be recovered, matches that were found were not erroneous. The results showed that there were many misses of correct matches, but very few false positive matches. Different variations of the algorithm achieved from 80% to 100% correct matches giving high confidence in the results of the method.

The data selection techniques were successful in improving the signal-to-noise ratio, though not significantly. Their main contribution was to reduce the data set in a consistent way. This essentially reduced the density of the data, making each item more unique and allowing more matches to be identified.

6.1 Future Work

There are many ways in which this work could be furthered.

To avoid the assumption of Lambertian surfaces, colour images could be used instead of black & white images. Then the colour hue and possibly saturation could be matched, ignoring the intensity which will change with the amount of reflected light.

The structure used for matching was four coplanar lines that intersected at a common point. The detection of this configuration was extremely difficult and many non-coplanar sets of lines resulted in invalid structures. This was the major problem encountered. Choosing a structure that could be detected more easily would improve the usefulness of the method.

As well, methods which further reduce the density of the data while retaining some consistent subset may be employed to ensure the method's functionality – in essence, data selection techniques which produce more data reduction. Choosing an invariant structure that occurs less frequently would produce a similar result and may be sufficient for improving the method. Using the invariant to classify a larger structure that can be found in a scene may also be fruitful further study.

However, the work has suggested what the author feels to be the most fruitful direction of future study. Instead of matching many small features, the largest object or objects that can be modeled should be located in each image. Ideally, these would be from a database of simple parameterized models (e.g. cube, cylinder). These very few objects could then be matched between images. An estimate of pose could immediately be calculated. Smaller features such as lines could then be matched between images, always relative to the large objects originally identified.

Bibliography

[Ballard82] Dana H. Ballard and Christopher M. Brown, **Computer Vision**, Prentice-Hall, 1982.

[Barnard83] S. T. Barnard, *Interpreting Perspective Images*, **Artificial Intelligence**, Vol. 21, 1983, pp. 435-462.

[Barrett91] Eamon B. Barrett, Paul M. Payton, Nils N. Haag and Michael H. Brill, *General Methods for Determining Projective Invariants in Imagery*, **CVGIP: Image Understanding**, Vol. 53, No. 1, January 1991, pp. 46-65.

[Beliveau90] Yvan J. Beliveau, Sankar Jayaram, Steven D. Johnson, *Reverse Engineering of a Product Model (RPM)*, **Close-Range Photogrammetry Meets Machines Vision**, SPIE, Vol. 1395, 1990, pp. 894-899.

[Burns86] J. Brian Burns, Allen R. Hanson, Edward M. Riseman, *Extracting Straight Lines*, **IEEE Trans. on PAMI**, Vol. PAMI-8, No. 4, July 1986, pp. 425-455.

[Coelho92] Christopher Coelho, Aaron Heller, Joseph L. Mundy, David A. Forsyth and Andrew Zisserman, *An Experimental Evaluation of Projective Invariants*, **Geometric Invariance in Computer Vision**, editors J. L. Mundy and A. Zisserman. MIT Press, 1992, pp. 87-104

[Dhond89] Umesh R. Dhond and J. K. Aggarwal, *Structure from Stereo*, **IEEE Transactions on Systems, Man, and Cybernetics**, Vol. 19, No. 6, November/December 1989, pp. 1489-1509.

[Eos92] advertising flyer, *Photo Modeler*, **Eos Systems, Inc.**, 205-2034 West 12th Avenue, Vancouver BC V6J 2G2, tel: 604-732-6658, http://www.wimsey.com/PhotoModeler/.

[Faugeras87] Olivier D. Faugeras, Francis Lustman, Giogio Toscani *Motion and Structure from Motion from Point and Line Matches*, **IEEE First International Conference on Computer Vision**, 1987, pp. 25-34.

[Faugeras93] Oliveier D. Faugeras, **Three Dimensional Computer Vision: A Geometric Viewpoint**, MIT Press, 1993.

[Forsyth90] David Forsyth, Joseph L. Mundy, Andrew Zisserman, Christopher M. Brown, *Invarianve - A New Framework for Vision*, **3rd International Conference on Computer Visions, Proceedings**, 1990, pp. 598-605.

[Gonzalez87] Rafael C. Gonzalez, Paul Wintz, **Digital Image Processing**, Addison Wesley, 1987.

[Granshaw80] S.I. Granshaw, *Bundle Adjustment Methods in Engineering Photogrammetry*, **Photogrammetric Record**, 10(56), pp. 181-207, October 1980.

[Hartley92] Richard I. Hartley, *Estimation of Relative Camera Positions for Uncalibrated Cameras*, **ECCV'92**, pp. 579-587.

[Hartley93] R. I. Hartley and J. L. Mundy, *The relationship between photogrammetry and computer vision*, **SPIE- Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision**, 14-15 April, 1993, Orlando, eds/chairs, Eamon B Barrett, David M. McKeown, Jr., Vol 1944, pp.92-105.

[Hartley94] Richard I. Hartley, *Projective Reconstruction from Line Correspondences*, **Proceedings 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition** (Seattle, Washington), pp. 903-907.

[Hartley95] Richard I. Hartley, *A linear method for reconstruction from lines and points*, **Proceedings of the IEEE**, 1995, pp. 882-887.

[Hecker91] Yaron C. Hecker and Ruud M. Bolle, *Invariant feature matching in parameter space with application to line features*, **Proceeding of SPIE - The International Society for Optical Engineering**, V1570, 1991, pp. 298-314.

[Hough62] P. V. C. Hough, *Methods and Means for Recognizing Complex patterns*, **U.S Patent 3,069,654**, 1962.

[Hussien93] B. Hussien and B. Sridhar, *A Robust Line Extraction and Matching Algorithm*, **Proceedings of SPIE - The International Society for Optical Engineering**, Vol. 2055, 1993, pp. 369-380.

[Jezouin90] J. L. Jezouin and N. Ayache, *3D Structure from a Monocular Sequence of Images*, **IEEE 3rd International Conference on Computer Vision**, 1990, pp. 441-445.

[Kittler93] A. Tai, J. Kittler, M. Petrou, and T. Windeatt, *Vanishing Point Detection*, **Image and Vision Computing**, Vol. 11, No. 4, May 1993. pp.240-245.

[Kumar94] Rakesh Kumar and Allen R. Hanson, *Robust Methods for Estimating Pose and a Sensitivity Analysis*, **CVGIP: Image Understanding**, Vol. 60, No. 3, November 1994, pp. 313-342

[Lee90] Chia-Hoang Lee and Thomas Huang, *Finding Point Correspondences and Determining Motion of a Rigid Object from Two Weak Perspective View*", **CVGIP**, No. 52, pp. 309-327, 1990

[Leung96] John C.H. Leung, Gerard F. McLean, *Vanishing Point Matching*, accepted for **International Conference on Image Processing '96**, Sept 16-19, 1996, Lausanne, Switzerland.

[Liu91] Y. Liu and T. S. Huang, *Determining Straight Line Correspondences From Intensity Images*, **Pattern Recognition**, Vol. 24, No. 6, 1991, pp. 489-504.

[Lowe85] David G. Lowe, **Perceptual Organization and Visual Recognition**, Kluwar Academic publishers, 1985.

[Magee84] M. J. Magee and J. K. Aggarwal, *Determining Vanishing Points from Perspective Image*, **Computer Vision, Graphics, and Image Processing**, Vol. 26, 1984. pp. 256-267.

[McLean96] G. F. McLean, *Geometric Correction of Digitized Art*, **Graphical Models and Image Processing**, Vol 58, No. 2, March 1996, pp. 142-154.

[Mohr95] Roger Mohr, *Projective Geometry and Computer Vision*, **Handbook of Pattern Recognition and Computer Vision**, draft, to be published by World Scientific publishing Company, eds. C. H. Chen, L. F. Pau and P. S. P. Wang, 1995, Chapter 1. Obtained from the author by Dr. Ged McLean.

[Mundy92] Joseph L. Mundy and Andrew Zisserman, *Appendix - Projective Geometry for Machine Vision*, **Geometric Invariance in Computer Vision**, eds. Mundy and Zisserman, MIT Press, 1992, pp. 463-519.

[Nacken93] Peter F. M. Nacken, *A metric for Line Segments*, **IEEE Transaction on Patterm Analysis and Machine Intelligence**, Vol. 15, No. 12, December 1993, pp. 1312-1318.

[Petsa94] Elli Petsa and Petros Patias. *Relative Orientation of Image Triples Using Straight Linear Features*, **Spatial Information from Digital Photogrammetry and Computer Vision**, Sept 5-9, 1994, Munich, Germany, SPIE, Vol. 2357, pp.663-669.

[Prescott94] B. Prescott, **An Optimization Approach to the Correction of Radial Distortion for Calibration for Digital Cameras**, Master's thesis, Department of Mechanical Engineering, University of Victoria, Victoria, British Columbia, Canada, 1994.

[Quan88] Long Quan and Roger Mohr, *Matching Perspective Images Using Geometric Constraints and Perceptual Grouping*, **IEEE 2nd International Conference on Computer Vision**, 1988, pp. 679-683

[Semple52] J. G. Semple, G. T. Kneebone, **Algebraic Projective Geometry**, Oxford University Press, 1952.

[Schalkoff92] R. Schalkoff, **Pattern Recognition**, John Wiley & Sons Inc., 1992.

[Shigang90] Li Shigang, Saburo Tsuji and Masakazu Imai, *Determing of Camera Rotation from Vanishing Points of Lines on Horizontal Planes*, **3rd International Conference on Computer Vision**, 1990, pp. 499-502.

[Tsai86] Roger Y. Tsai, *An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision*, **Proceedings of IEEE Conference on Computer Vision and Pattern Recognition**, Miami Beach, FL, 1986, pages 364-374.

[Tsai87] Roger Y. Tsai, *A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*, **IEEE Journal of Robotics and Automation**, Vol. RA-3, No. 4, August 1987, pages 323-344.

[Warring96] Marcel Warring, *Digitized Circular Arcs: Characterization and Parameter Estimation*, **IEEE Trans on PAMI**, Vol. 18, No. 2, Feb. 1996, pp. 155-160.

[Weng92] Juyang Weng, Thomas S. Huang, Narendra Ahuja, *Motion and Strurcture from Line Correspondences: Closed-Form Solution, Uniqueness, and Optimization*, **IEEE Trans on PAMI**, Vol. 14, No. 3, March 1992, pp. 318-336.

[Wolf83] Paul R. Wolf, **Elements of Photogrammetry**, 2nd edition, McGraw Hill, 1992.

Vita

Surname: Vanderkooy

Given Names: Geoffrey Edward

Place of Birth: Cambridge, England

Educational Institutions Attended:

University of Victoria	1994 to 1996
University of Waterloo	1986 to 1991

Degrees Awarded:

B.A.Sc., University of Waterloo	1991

Honours and Awards:

Advanced Systems Institute GRAP Scholarship	1994
President's Research Scholarship	1994
NSERC Scholarship for Graduate Studies	1991
Work-term Report Award	1990
Waterloo County Entrance Scholarship	1986
Ontario Provincial Scholar	1986

Publications:

Partial Copyright License

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of the thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis:

Line Correspondence for Unknown Camera Positions

Author [redacted]

Geoffrey Edward Vanderkooy

June 18, 1996