2001

AUTOMATIC IMAGE TO MODEL ALIGNMENT FOR PHOTO-REALISTIC URBAN MODEL RECONSTRUCTION

Mike Partington
University of Kentucky, lexpart@juno.com

ABSTRACT OF THESIS

AUTOMATIC IMAGE TO MODEL ALIGNMENT FOR PHOTO-REALISTIC URBAN MODEL RECONSTRUCTION

We introduce a hybrid approach in which images of an urban scene are automatically aligned with a base geometry of the scene to determine model-relative external camera parameters. The algorithm takes as input a model of the scene and images with approximate external camera parameters and aligns the images to the model by extracting the facades from the images and aligning the facades with the model by minimizing over a multivariate objective function. The resulting image-pose pairs can be used to render photo-realistic views of the model via texture mapping.

Several natural extensions to the base hybrid reconstruction technique are also introduced. These extensions, which include vanishing point based calibration refinement and video stream based reconstruction, increase the accuracy of the base algorithm, reduce the amount of data that must be provided by the user as input to the algorithm, and provide a mechanism for automatically calibrating a large set of images for post processing steps such as automatic model enhancement and fly-through model visualization.

Traditionally, photo-realistic urban reconstruction has been approached from purely image-based or model-based approaches. Recently, research has been conducted on hybrid approaches, which combine the use of images and models. Such approaches typically require user assistance for camera calibration. Our approach is an improvement over these methods because it does not require user assistance for camera calibration.

Keywords: Urban Model Reconstruction, Facade Extraction, Vanishing Points, Photo-realistic Models, Image to Model Alignment.

Mike Partington

May 4, 2001

AUTOMATIC IMAGE TO MODEL ALIGNMENT FOR PHOTO-REALISTIC URBAN MODEL RECONSTRUCTION

By

Mike Partington

Christopher O. Jaynes
Director of Thesis

Grzegorz Wasilkowski
Director of Graduate Studies

May 4, 2001

RULES FOR THE USE OF THE THESES

THESIS

Mike Partington

The Graduate School

University of Kentucky

2001

AUTOMATIC IMAGE TO MODEL ALIGNMENT FOR PHOTO-REALISTIC URBAN MODEL RECONSTRUCTION

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
at the University of Kentucky

By

Mike Partington

Lexington, Kentucky

Director: Dr. Christopher O. Jaynes, Professor of Computer Science

Lexington, Kentucky

2001

To Julie, whose love and sacrifice made this possible.

ACKNOWLEDGMENTS

The following thesis was inspired to a great extent by the leadership and enthusiasm of my Thesis Chair, Dr. Christopher Jaynes. Under his direction, what started as a class project grew into an exciting research topic that has consumed much of my life over the past two years. In addition, I wish to thank the faculty of the University of Kentucky's College of Engineering for providing the opportunity to gain a solid educational foundation.

I am forever grateful to my parents, who taught me the importance of education by precept and example from an early age. The completion of a Master's degree is the culmination of goals that they helped me to establish many years ago. Although this goal is accomplished, the yearning they placed in my heart for knowledge and understanding is not quenched, but grows ever stronger.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FILES

partington_thesis.pdf (PDF file, 1.8 MB)

Chapter 1
Introduction

An active area of research in computer vision is the automatic reconstruction of 3D urban models from imagery. Such models are useful in a number of important tasks such as city planning, disaster route planning, and traffic management. Accurately rendered models can support advanced visualization tasks such as site walk/fly-throughs and mission planning. An automated system that is capable of providing geometric as well as image-based 3D models of urban regions reduces the time and cost required by human assisted approaches to creating such models. As a consequence, there has been a focus on accurate reconstruction of urban models using a variety of techniques.

Due to the large coverage provided by high altitude and space borne sensors, as well as observable building boundaries, a significant amount of research has been focused on reconstruction of sites using aerial images [1][2][3][4][5][6]. Approaches range from active radar such as interferrometric synthetic aperture radar (IFSAR) [7], to fully passive techniques [3][4]. These approaches have made dramatic progress in the previous ten years and are now reporting building reconstruction accuracies of over 95% for reasonably complex datasets [7][8].

A significant drawback of these approaches, however, is that they focus on the acquisition of a three-dimensional model (typically a building footprint and rooftop shape) that is observable from aerial data only. Although approaches to detecting windows, doors, and other model substructure from aerial images have been suggested [9][10], the final model is restricted to resolutions limited by the aerial sensors and the extreme foreshortening of the building facade based on the viewing position of the sensor. Even images taken at oblique angles will most likely not reveal a significant building façade due to occlusion by other buildings and structures located at the site. In addition, structural details will be beyond the resolution of many aerial sensors. These approaches, while very useful for accurate reconstruction of a site's base-geometry, do not address the incorporation of high-resolution, ground level imagery that may be available for the site in question.

In contrast to this approach, other researchers have focused on reconstruction of urban scenes exclusively from ground level imagery [11][12]. Ground level approaches use images taken at relatively close range to the urban structures and then use these images for tasks such as model reconstruction and view interpolation. Various approaches have developed for varying types of input including single snapshots and video sequences with varying amounts of camera calibration information [11][12][13][14]. Typically in ground level approaches a representation of a scene must be pieced together from multiple images. This need to combine information from various sources results in several complexities such as multi-image structural correlation and resolving overlapping textures. However, due to the close range nature of the photography, ground level approaches can yield detailed and photo-realistically accurate models and views of urban structures.

Of the many such approaches that have been investigated, the most promising determine relative camera transformation parameters and use this information to construct a 3D model using feature correspondences and epipolar geometry [11][13][14][15]. However, although capable of accurately reconstructing point locations on a structure, these approaches regularly yield uneven surfaces that do not render realistically from arbitrary views. This is due to error in the derived point locations to which model surfaces must be fitted. Furthermore, techniques based on feature correlation may perform poorly in the presence of occlusions or other factors that change the feature point characteristics between images.

The goal of the research presented in this thesis is to produce *photo-realistic* models of buildings in urban environments. *Photo-realistic* models are perceptually identical to a photograph of a building. Achieving photo-realism requires that models are capable of representing accurate surface substructure (windows, doors, alcoves), surface texture (bricks, wood siding, stucco), and radiometric properties (color, reflectance). Models that exhibit photo-realistic characteristics are desirable for uses such as architectural analysis, real-estate commerce, virtual tourism, and entertainment.

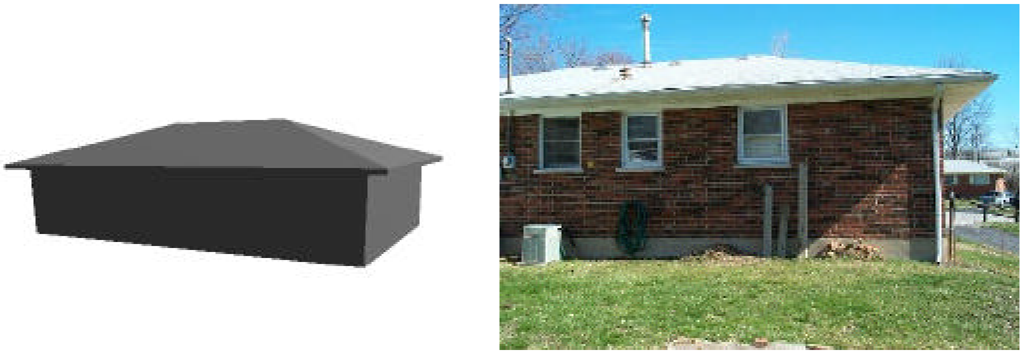
Techniques for creating photo-realistic models fall into two categories. Model-based approaches attempt to create a photo-realistic effect by creating a geometrically detailed model. These approaches define significant (i.e. likely to be observable from the expected set of viewing angles) physical and visual aspects of a structure using automated or semi-automated methods. For example, model-based approaches may produce a distinct modular component for each brick in a wall, for each pane of glass in a window, and for each piece of trim or gutter on a façade. Each of these model components are typically assigned radiometric and material properties that render a realistic effect.

Researchers have focused on the different technical challenges related to automatic model acquisition for several decades. Approaches include shape from shading [16][17][18], feature detection and grouping [9][19], surface fitting [20], surface estimation [21][22][23], and color and reflectance computation [24][25][26][27][28]. Although this type of explicit geometric description of the scene leads to high resolution and often visually accurate imagery, these approaches are prohibitively costly for wide scale applications. They also typically require a significant amount of user assistance in order to create a model that is sufficiently detailed to provide a photo-realistic appearance. The advantage of the model-based approach, however, is that an explicit model is provided by the reconstruction system. Simulations, geometric and material databases, and other applications may require the details afforded by an explicit three-dimensional site model [29][30].

On the other hand, image-based approaches seek to produce a set of visually accurate views of a scene directly from a set of images without the need for an explicit geometric representation [11][12][14][31]. Generally speaking, these techniques rely upon the existence of precise, known camera parameters in order to effectively interpolate views directly in projective space. Some image-based techniques assume that such information is available with image sets [12]. Others attempt to estimate camera parameters based on urban characteristics or image feature correlation techniques [32][33][34][35]. In any case, image-based approaches suffer from the problem of having to interpolate views from existing data. Effectively interpolating views, particularly of arbitrary, non-planar surfaces, can be very difficult; however, several techniques have been developed to address this problem [11][12][36]. In spite of these approaches, in any given sequence set of images of a scene, there is a high probability that some portion of the scene structures will not be observed in any of the images. If this portion of the structure is then viewed from a novel location, interpolation will most likely fail to produce a coherent, realistic scene.

The approach presented in this thesis is a hybrid approach. The technique is driven by the observation that aerial data is a rich source of information for the base geometry of a site, while incorporated ground-level imagery is important for achieving a complete photo-realistic and high resolution model. Ground level imagery, if aligned to a base model, can be used for effective image interpolation. As demonstrated by Debevec [36], if a set of images of a scene are available, then a rendering of the scene can be constructed by interpolating between images whose camera parameters are near the viewing position. The value of a rendered pixel at a given model location can be interpolated by using a weighted average of near images based on proximity to the viewing position. Alternatively, a non interpolative technique can be used in instances where a large number of views are available by utilizing the observation that an image projected onto a structure will yield an accurate visualization of the structure within a locus about the camera location. As a viewer moves through a scene, the image most nearly matching the current viewing parameters can be loaded and projected onto the base model to yield an accurate representation from the current viewing location. As the viewer moves, other images are loaded and projected to continue to provide an accurate visualization at new viewing locations. In order to realize such a system, we need a method for acquiring the image-pose pairs used by the algorithm. Hence, an automated algorithm that is capable of determining accurate frame-by-frame camera parameters is desirable. The hybrid approach presented here is such an algorithm.

The approach presented in this thesis assumes that a base model is acquired initially using techniques introduced by [1][7]. Then, ground level images of the scene are automatically aligned to the model by extracting the facades from each image and minimizing the alignment error between the facade and the model. The data required by this approach is a simple model (see Section 2.4.1) of a structure or group of structures, images of the structures, the approximate extrinsic camera parameters for each image (see Section 2.4.2), and the internal parameters of the camera. Figure 1 depicts the type of data used by the system. Building boundaries and their three-dimensional outline are derived using aerial or other techniques such as straightforward by-hand modeling of the significant building boundaries (see Figure 1a). Ground level images captured at the site are then aligned to the model (Figure 1b).



(a) (b)

Figure 1. (a) Rendered 3D model used as input to the alignment algorithm. (b) Ground level image of the structure.

Once a three-dimensional model of the site is available, the approach proceeds in two phases for each image (see Figure 2). The first phase extracts façades from the image using a vanishing point analysis technique. The second phase aligns extracted façades with the base model by automatically estimating the camera parameters that align facade features in the image with those on the model. This estimation is performed by minimizing over an objective function which measures the fit between a particular image-pose pair and the model. The output is camera parameters for each image that correspond to the minimum value of the fitting error function between the extracted facades and the model. This information can then be used to project the image onto the model or for accurate rendering from a wide range of views and for post-processing techniques such as multi-image registration and model enhancement. Hybrid rendering approaches can exploit the set of image-pose pairs to further refine pose estimations and to sequentially process video sequences (see Appendix A).

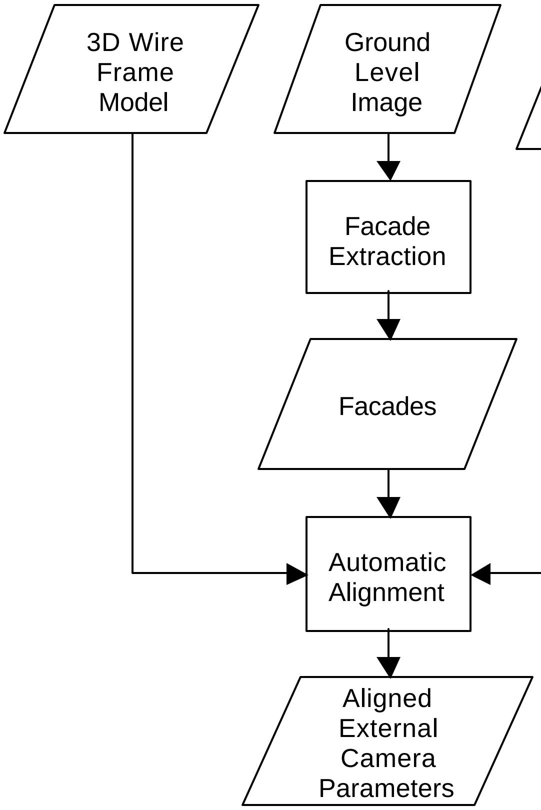


Figure 2. Flow of the hybrid approach.

The data created by the algorithm can also be used to create a photo-realistic environment without projecting the images onto a model provided that a sufficient number of images of a scene are available and the potential viewing positions of an agent are bounded. A given image/pose pair will be accurate within some neighborhood about the actual camera parameters. As an agent moves through a scene, the appropriate image can be loaded from an image database based on the agent's current viewing parameters, i.e. the image with the pose that most closely matches the agent's viewing parameters. Provided that enough image/pose pairs are available, the model can be explored by users who require a photo-realistic experience. A disadvantage of this approach is the limited allowed mobility of the agent. The photo-realistic effect will occur only while the agent is moving within the bounds of the parameter space of the image database.

Due to the dense coverage of camera parameter space provided by video sequences, video is a good medium for obtaining data for use by this technique.

1.1 Thesis

We assert that reconstruction of photo-realistic urban models is improved in both quality and in equipment and user cost by the automatic alignment approach described above. An integral part of this automatic alignment algorithm is the accurate extraction of building facades from images. We assert that facades can be accurately extracted from urban imagery by utilizing vanishing points. This assertion is based on the observation that facades typically contain a high degree of parallel structure relative to non-urban scenery (see Section 3.2). Other than ground level image acquisition, the approach described herein is completely automated.

This thesis will show that the hybrid approach is a viable method for extracting façades and automatically aligning images to models under the following conditions:

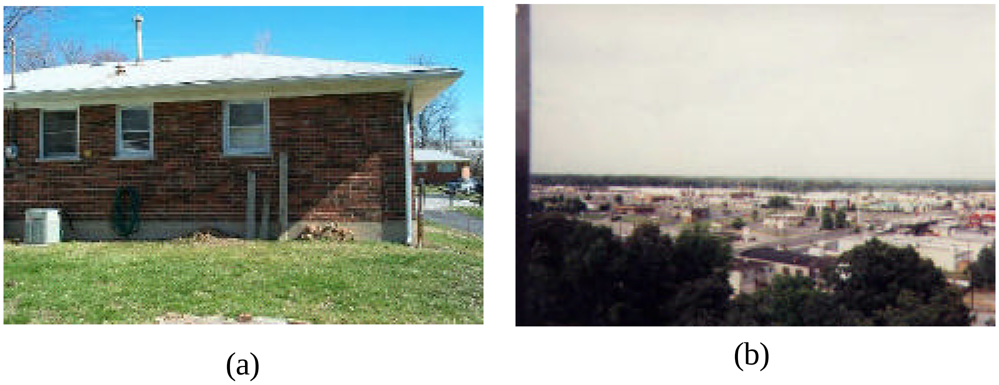
1 Input images are primarily dominated by building facades.

2 The photographed facades contain significant features that are parallel in the world. In particular, the rooftop should be straight and parallel to other facade features.

3 Façade edges are straight and parallel.

Condition (1) is required because the façade extraction algorithm assumes that groups of lines that share a vanishing point arise from building facades. If the percentage of image space occupied by facades is not sufficiently high, then the algorithm may not be able recognize the vanishing points induced by the facades. Condition (2) is also required for the façade extraction algorithm to correctly identify vanishing points. The algorithm identifies vanishing points by finding clusters of line intersections. If facades do not contain significant parallel structure, then prominent intersection clusters will not be formed. Condition (3) is needed in order for the image to model alignment minimization routine to correctly align images to the model. The alignment algorithm attempts to align extracted facades to façade edges in the model. If these edges are not represented in the extracted façade lines, then the minimization routine will produce an incorrect alignment. Façade edges that are straight and parallel to other surface features should be extracted by the façade extraction algorithm, thus maximizing the success of our approach.

Figure 3 shows two images of building facades. Image (a) represents an ideal input image for this algorithm while image (b) represents an image that is ill-suited for this algorithm. In image (a), for each face that is visible, the boundary lines of the face are also visible, providing the alignment algorithm with a good chance of extracting façade edges that are represented in the model. In addition, opposing boundary lines for the faces are visible, meaning that the entire width or length of the face is included in the image. Although image (b) contains several buildings, facade structure does not dominate the image. Rather, individual buildings occupy only small portions of the image area. In addition, the image is not sufficiently close to any particular building to provide photo-realistic quality data for a synthesized near view. The structures that are close to the camera are occluded by trees. Finally, the grainy texture of the image will decrease the ability of the line finder to accurately detect edges.



(a) (b)

Figure 3. (a) An example of an image that is well suited for the automatic alignment algorithm, and (b) one that is ill-suited for the algorithm.

1.2 Photo-realistic model reconstruction difficulties

Automatic and semi-automatic photo-realistic model reconstruction is difficult for several reasons. Model-based approaches are dependent on the accuracy of the model and the ability of a rendering algorithm to create visually convincing views. Creating such models is prohibitively expensive for large scale applications. On the other hand, purely image based approaches are dependent on the accuracy of the provided or calculated pose information. One approach to ensuring that pose information is highly accurate is to use precisely calibrated equipment so that both intrinsic and extrinsic camera parameters are known [12]. This calibration equipment can be both expensive and cumbersome to operate. Other approaches automatically correlate images by finding image to image correspondences and estimating pose information [13][14][31]. These automated approaches are difficult to implement due to scene variations between viewing angles such as lighting differences, occlusions, and variations in surface textures. Yet another approach is to estimate pose information by using invariant geometric properties exhibited by urban structures [1][13][32]. In contrast, the approach described in this thesis does not require multiple, overlapping views of the building to be acquired. Though multiple views can be exploited for increased performance (see Appendix A), the fundamental algorithm is designed for a single view.

The hybrid approach reduces image correspondence problems by assuming the existence of a base model. However, whether a model is acquired manually or automatically through means such as aerial reconstruction, the model will contain some amount of error which may affect the outcome of the image to model alignment. It is also difficult to align ground-truth images to the model. Construction of an objective function that accurately computes the degree to which an image/pose pair align with a model is subject to the same correspondence problem as other automated registration techniques.

Regardless of whether the hybrid approach or a purely image based approach is used, if the estimated pose parameters are not accurate, then the resulting view reconstructions will not be convincing. Consider an example of the hybrid approach in which an image is projected onto a model. Figure 4 (a) shows a model and the location of a camera as measured at image acquisition time. The model, referred to as Model M, was acquired by hand measurements of the structure and is accurate to within 10 cm at any vertex. Figure 4 (b) is an 1440x960 pixel image of the structure represented by Model M taken at the camera location. This image will be referred to as image A throughout the subsequent discussion. When the image is projected onto Model M using the exact camera parameters, the model is given a photo-realistic appearance as shown in Figure 5 (a). If the camera parameters are in error by only a small amount, then the resulting projection loses its realistic appearance. Figure 5 (b) shows the model when the image is projected from a slightly different location than the actual image location.

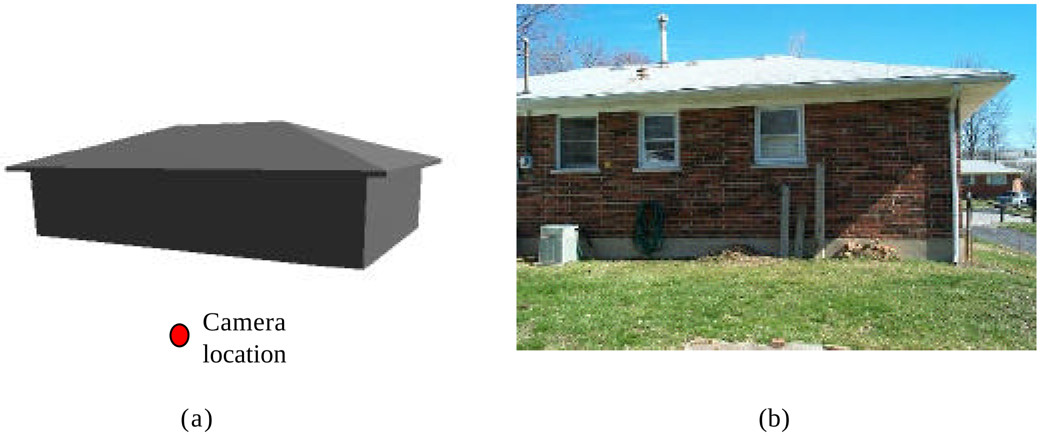


Figure 4. (a) Model M. Model of house with measured camera location. (b) Image A. Image of structure at the camera location.

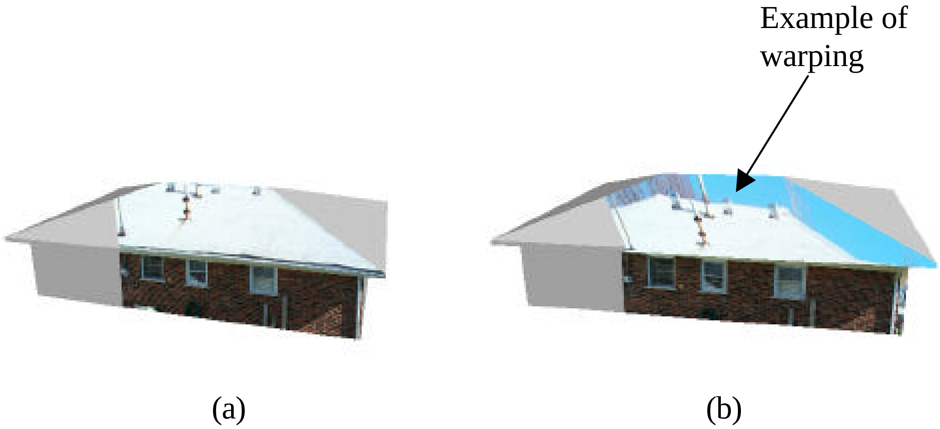


Figure 5. (a) Novel view of Model M with Image A projected from the measured camera location. (b) Novel view of Model M with Image A projected with perturbed camera parameters. The image was projected from a location with 0.3 m of error in two translation parameters and 1 degree of error in a rotational parameter.

To further complicate matters, a single image of a structure, when projected onto a model of the structure, yields a representation that is accurate only when viewed from the camera location. Consider again Figure 5a in which an aligned image/pose pair are used to project the image onto the model. When the same projection is viewed from a different location, as shown in Figure 6, features of the model, such as the roof vents and chimney become distorted. This phenomenon is caused by unmodelled structure in the image. We assume that façade surfaces are planar and do not compensate for nonplanar structure. Rather, we argue that, as explained above, we can somewhat compensate for unmodelled structure by acquiring a large set of images from a variety of viewing locations and load the image that is most appropriate for the current viewer parameters. The effectiveness of this technique is dependent on both the amount and shape of the unmodelled structure and on the distance between neighboring images. Image-based approaches also suffer from image illumination variance. Shadow and lighting differences for different cameras can produce conflicting values for the characteristics of a given surface location. These differences can be compensated for using techniques developed for image interpolation and mosaicing [11][36].

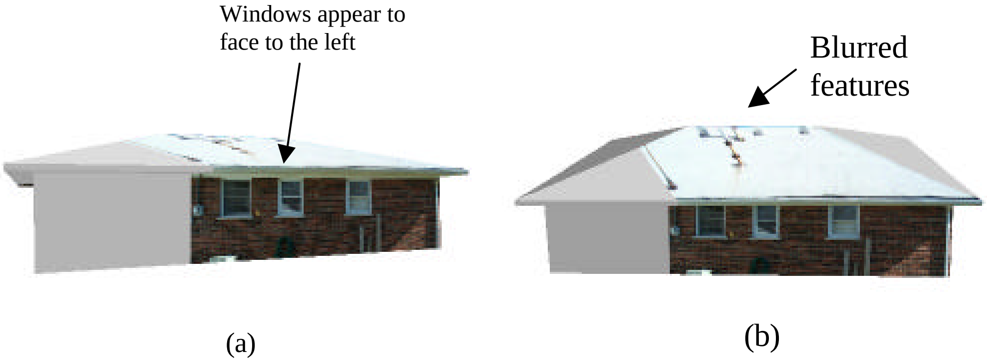


Figure 6. Model M viewed from novel positions that show how features can appear distorted when a single image is projected and viewed from a vastly different location. Note that the vertical features on the roof are blurred and the windows appear to face the wrong way.

Chapter 2
Preliminaries and Related Work

Although the automated, hybrid approach presented in this thesis has not been explored previously, it is built upon a long research history in various areas within computer vision. In this section we introduce several of the key concepts underlying the hybrid approach and the work most closely related to each.

2.1.1 Vanishing Point Extraction

A key step in automatic alignment of imagery to the approximately planar structure found in urban environments is the detection of significant vanishing points in the scene. Vanishing point extraction from 2D imagery has been a topic of research for many years [37][38][39][40]. Most approaches are based on intersecting lines in the image and searching for concentrations of intersections. The most commonly used approach, as explained by Barnard [37], utilizes a Gaussian sphere to bound the image line intersection space. In Barnard's method, image lines are projected from the camera center through a Gaussian sphere to form great circles on the sphere. Clusters of intersections of great circles correspond to vanishing points. The clusters are found on the sphere by a histogramming technique, or other methods [40]. In addition to the Gaussian sphere approach, various image space approaches for finding vanishing points have also been explored [39][41]. Most of the proposed vanishing point extraction techniques are hampered by the need to bound and equally divide the intersection space of image lines.

Our approach is an image space approach in which lines are extended and intersected. Intersections are then grouped to detect significant intersections that arise from image structure. However, rather than use a histogramming technique in which the image space is segmented into roughly equal regions, we base our intersection grouping method on an analysis of the expected intersection errors. Intersections are grouped by calculating whether the intersections lie within the region bounded by the expected error in an intersections' locations. Although computationally complex, the technique avoids the need to develop an image space segmentation algorithm and results in very accurate groupings of intersections (see Section 4.2.2.3).

2.1.2 3D reconstruction from vanishing points

Once lines are known to share a vanishing point in image space, probabilistically, we assume that the lines are parallel in world space and can be used to compute 3D characteristics. For example, if two lines in the image are known to be parallel in the world, then the direction cosines of the world lines can be found using perspective projective geometry characteristics [42]. Such information can be used to refine approximations and to test hypothesis about the structure in a scene. Shufelt uses assumptions about image objects to enhance the accuracy of the vanishing point extraction process and to more accurately reconstruct buildings from aerial imagery [41]. Rather than searching simply for vanishing points to indicate the presence of manmade structure, he searches for groups of vanishing points that are consistent with expected structures in the scene. For example, rectangular objects, when viewed from aerial perspective, are expected to produce two vanishing points that are related by the relative orientation between the ground plane normal and the camera normal. Utilizing this information allows for more accurate object extraction in aerial photography. Teller uses multiple images of urban scenes to perform multi-image registration for the purpose of refining approximated camera parameters. He utilizes the vanishing points induced by the images for this registration step. He then uses the registered and calibrated images to reconstruct scene objects in 3D [32].

2.1.3 Aerial reconstruction research

The technique presented here also relies upon the existence of an accurate base geometry of the subject structure or group of structures. A source of such base models is 3D aerial reconstruction, a field which has been explored intensively since the early 1980's. Although several aerial reconstruction systems have been developed that are capable of reconstructing models of various types of buildings with high accuracy [4][5][6][7][30][43], reliance upon aerial reconstruction techniques for base models introduces some restrictions on the type of urban structures that can be used in the hybrid approach. These restrictions are introduced by the ability of current aerial reconstruction systems to accurately reconstruct buildings of only certain types.

Early aerial reconstruction techniques were only able to construct 2D models of buildings [44][45], which would be wholly unsuitable for the hybrid approach. As technology advanced, 3D reconstruction became possible for restricted building shapes and roof types [8][46]. More recently, aerial reconstruction systems have advanced to the point of being able to accurately extract buildings with complicated footprints and with roof types of various parametric classes [6][7]. These recent techniques have the ability to provide models of sufficient accuracy for our automatic image to model alignment approach. However, until aerial reconstruction techniques can conquer the roof complexities and complex footprint types that occur in some buildings, other methods will have to be used on these structures for obtaining models suitable as input to the automatic alignment algorithm.

2.1.4 Ground level reconstruction

Ground level reconstruction is currently an active area in computer vision research. Most techniques are based on multi-image registration and analysis [47]. However, approaches vary in how they register the images. Some techniques assume that the exact camera parameters are provided with the input images [12]. Others attempt to determine these parameters through feature correlation [11][15][35][48] or vanishing point techniques [32]. Pollefeys uses image correspondences to perform auto calibration of camera parameters in reconstructing objects of varied shape and size [14]. He assumes that image to image camera parameter variation is relatively small and uses this information to improve feature correlation and guess camera locations. Stereo reconstruction techniques are then used to reconstruct the scene's 3D structure from image correspondences. The process iteratively improves the reconstruction as each image in the sequence is processed. Similarly, Beardsley uses feature correspondences in a sequence of video images to determine relative camera calibration parameters over groupings of three consecutive images [31]. This process also iteratively improves an estimated 3D model as each additional image is analyzed. Faugeras uses feature correspondences to calibrate cameras but enlists the help of the user for 3D reconstruction [11]. After the cameras are calibrated, the user selects sets of parallel lines in the image, which are then used to determine information such as surface normals. Finally, the model is given a photo-realistic appearance by utilizing mosaicing techniques. Teller approaches the problem by constructing spherical mosaics using highly calibrated and dense imagery at tightly spaced nodes in a scene. He relaxes camera calibration requirements somewhat by aligning vanishing points between nodes. He then reconstructs scene geometry from feature correspondences and the abundance of calibrated imagery using various techniques developed over the last decade.

An approach particularly related to the hybrid approach presented in this thesis was developed by Debevec [13]. His semi-automated approach builds primitive models of a structure with the help of the user. The user selects lines in the image set and fits parametric structures to the selected lines to create an initial model. This model is then used to further refine the alignment between model and image. However, the technique relies upon the user to select line correspondences between images and models, which greatly simplifies the image to model alignment problem. We seek to eliminate user assistance altogether.

2.1.5 Facade extraction

Although there is not a large body of research available on urban facade extraction, there are several reconstruction algorithms that extract facades from images during the 3D reconstruction process. Wang utilizes aerial techniques to extract facades from an image [9][10]. The technique extracts facades by locating rooftops and identifying adjacent regions that match expected facade characteristics. The results of this effort are hampered by the extremely foreshortened nature of aerial data and poor resolution of fine structure at distant viewing locations. Teller extracts facades by searching for dense regions of appropriately oriented lines [33]. The technique is similar to the “plane-sweep” technique introduced by Collins [1] for detecting planar rooftops. The approach relies on the availability of well calibrated imagery to predict the orientation of horizontal lines in a scene. After identifying horizontal lines, it is relatively safe to conclude that a region of space occupied densely by horizontal lines corresponds to a facade in the context of an urban environment. Although very accurate, this approach relies on precisely calibrated imagery. Our efforts differ from these approaches in that we utilize vanishing points exclusively for facade extraction with no assumptions about camera parameters.

2.2 Summary of Related Research

As discussed above, there are many areas of computer vision research in 3D reconstruction that relate to the hybrid approach. Table 1 summaries the research documented above by selecting a representative sample from recent 3D reconstruction contributions.

Table 1. Comparison of selected 3D reconstruction techniques.

Research	Year	Related Features
Beardsley, Torr, Zisserman	1996	Automated, image-based approach using video sequences. Use feature correlation to compute relative camera transformation matrices via trifocal tensor. No camera parameters are required. Estimates 3D structure with first image grouping and refines with each successive grouping.
Debevec, Taylor, Malik	1996	Semi-automated, hybrid approach using snapshots. User assists image correlation by selecting lines in an image to create a base geometry. Base geometry used for further image alignment again utilizing user selected lines. Requires internal camera parameters. Constructs novel views using view-dependent texture mapping.
Faugeras, Robert, Laveau	1997	Semi-automated, image-based approach using snapshots. Use feature correlation to compute relative camera transformation matrices. No camera parameters are required. User assists geometric reconstruction by selecting parallel lines. Textures are mapped onto the reconstructed geometry using mosaicing.
Collins, Jaynes, Cheng, Wang, Stolle, Schultz, Hanson, Riseman	1997	Automated approach using aerial imagery. Internal and external camera parameters are required. Reconstructs building footprints, heights, and rooftops via corner graph construction and model fitting. Projects image information onto base geometry to yield low resolution surface detail.
Pollefeys, Koch, Vergauwen, Deknuydt, Luc Van Gool	2000	Automated, image-based approach using video sequences. Uses feature correlation to compute relative camera transformation matrices. No camera parameters are required. Reconstructs structure after all images have been calibrated using stereoscopic triangulation techniques.
Teller	2000	Automated, image-based approach using hemispherical image nodes. Requires knowledge of internal camera parameters and estimate of external camera parameters. Uses vanishing points to refine external camera parameters. Computes coarse surface structure and texture maps this surface with image information. Finishes by adding relief map to surfaces.
Partington, Jaynes	2001	Automated, hybrid-approach using snapshots or video. Requires a base geometry and internal camera parameters and approximate external camera parameters for each snapshot or for the first image in a sequence. Uses vanishing points to extract facades and aligns the facades with the base geometry. Calculate pose information is used to texture map the model with image data.

2.3 Advantages of the hybrid approach

The hybrid approach described in this thesis for creating photo-realistic models is a step towards not only automating the model creation process, but also towards greatly simplifying it. The idea of the approach is to project ground truth images onto a simple model in order to give the model a realistic appearance. The method does not rely on human assistance or precisely calibrated imagery to align an image with a model. Rather, we use assumptions about how buildings look to programmatically determine precisely where an image of a building was taken. Hence, the amount of user assistance needed to create a photo-realistic model using the approach is greatly reduced over conventional approaches.

Automatic image to model alignment has several advantages over other approaches to photo-realistic model reconstruction. Provided that a building model is available, the required equipment -- a digital camera and a GPS sensor or tape measure -- is relatively low cost. The hybrid approach further reduces cost by decreasing the need for human assistance. The operator of the digital camera need only be aware of relatively simple guidelines for image acquisition in order to gather data (see Section 2.4.2). Although the operator must record the approximate location and orientation of the camera for each image, this pose information need only be accurate to within a few feet in location and a couple degrees in orientation (see Section 6.5), which can be accomplished using the GPS sensor, a tape measurer, or even educated guessing. Due to the simplicity of the data acquisition process, little training is needed for gathering data. Finally, because this process is automated, it opens the door to a whole host of automated reconstruction possibilities including automatic model enhancement and even automatic model creation.

2.4 Data Requirements

The input data for the automatic image to model alignment algorithm consists of image/pose estimation pairs and a basic model. Methods for acquiring a base model are beyond the scope of this paper and will not be discussed in detail. The chosen model creation method is significant only in its ability to generate a wire-frame model that is sufficiently accurate for the image to model alignment to converge to the correct aligned location. Even hand approaches are adequate if they can yield the needed accuracy.

2.4.1 Model Requirements

The model of the target building or group of buildings need only include façade base planes. The location of model vertices within the scene's coordinate system should be accurate to within a few inches. The model's accuracy is crucial during the error minimization stage of the automatic alignment algorithm. During this stage, image lines will be compared with model lines to determine how well the model "fits" the image. In addition to meeting basic accuracy requirements, it is helpful to anticipate which building features are likely to be extracted by the facade extractor and, if possible, to include some of these features in the model. In general, the more the number of correspondences between extracted façade data and model data, the greater the accuracy of the automatic alignment. Finally, when acquiring a model, because the model defines the coordinate system for camera location approximations, it is a good idea to choose model units that correspond to easily measured world units, such as feet or meters.

2.4.2 Image Capture & Pose Estimation Requirements

Several guidelines can be followed to help ensure that the image data of a scene will be suitable for our hybrid approach. First, the approximate position and orientation of the camera in model space coordinates should be recorded as each image is taken. This approximate position will be used as the starting point for the alignment algorithm and should be accurate to within two degrees in orientation and three to four feet in location as determined in chapter 6.5. Second, images should be taken in as high resolution as possible. Using high resolution images facilitates the facade extraction algorithm by providing it with detailed and accurate lines. Third, obstructions in the images should be kept to a minimum. Trees, fences, people, and other obstructions can hide crucial facade features, resulting in decreased building facade extraction performance. Finally, images should avoid face transition boundaries, i.e., locations where a face is barely visible or barely not visible. As explained in Section 5.3, locations where faces come in and out of view can correspond to discontinuities in the alignment objective function. Pragmatically, we have found that somewhat unconstrained imagery can be used by the system. For example, results in Chapter 6 show that users can be asked to “take several images of the building facades” with little or no other instructions. Rotational parameters must be accurate to within one to two degrees (see Section 6.5), which is within the accuracies that can be expected from differential GPS sensors or when fitting video sequences (see Appendix A).

2.5 Contributions

This thesis enhances the current body of research on photo-realistic urban model reconstruction. In particular, our research makes the following contributions:

1 a new, automated, hybrid approach for creating photo-realistic models of urban structures.

2 a new algorithm for robust and automatic extraction of building facades from vanishing points.

3 an extension to the hybrid approach for minimizing image-to-model alignment error based on aligning extracted vanishing points in camera space along a vanishing line. This technique is useful when multiple views are available.

This thesis does not attempt to address approaches for creating the base models needed as input to the hybrid approach. Aerial reconstruction is discussed as an example to show that such approaches exist and that the resulting models are widely available. The chosen approach is important only to the degree that it produces accurate results in terms of error in the reconstructed base geometries. Many approaches are available that constrain planimetric error to 0.2-0.3 meters and alimetric error to 0.3 meters, which satisfies the accuracy demands required by the hybrid approach. Because only a wire frame model is required, even simple, hand created models can serve as input to the algorithm.

2.6 Guide to the thesis

Chapter 1 summarizes the approach explained in this thesis and compares it to other approaches. The thesis statement itself is located in Section 1.1. Chapter 2 further explains the approach taken by this thesis and includes motivation. In particular, Section 3.2 explains the reasoning for using vanishing points to extract building facades. describes the façade extraction algorithm in detail, beginning with line segment extraction in Section 4.1 and ending with vanishing point group thinning in Section 4.2.3. Chapter 5 begins with a discussion of multivariate minimization and then explains how this technique is applied to the hybrid approach to automatically align image/pose pairs with base models. Our objective function for the minimization is defined in Section 5.2. The objective function's strengths and weaknesses are discussed in Section 5.3. Section 6.1 explains the experimental setup for evaluating the performance and viability of the hybrid approach. Chapter 6 also documents the results of running the experiments. Chapter 7 summarizes the conclusions that can be drawn from this research and suggests strategies for improving the approach. Finally, extensions of the hybrid approach to video, including temporal assisted alignment and error minimization using vanishing lines, as well as preliminary results from these extensions, are considered in Appendix A.

Chapter 3
Automatic Alignment Basics

The algorithm for automatic image to model alignment consists of two steps: building façade extraction, and model alignment. The success of model alignment is dependent on the quality of façade lines extracted by the façade extractor and the accuracy of the provided base model. During model alignment, the facade boundary lines are automatically aligned with the base model. If the alignment is successful, then the image can be projected onto the model to create a photo-realistic model of the building from viewpoints that neighbor the position of the acquired image. As stated earlier, it is assumed that a base model is provided via some 3D reconstruction technique (see Section 2.4). This section will discuss the building extraction and model alignment algorithms. Before discussing the details of these algorithms, it is helpful to review the basic principles involved in extracting objects, and in particular buildings, from an image.

3.1 Feature Extraction

When searching for particular feature in an image it is necessary to define the invariant characteristics of the information. In particular, features that are invariant to the set of transforms that the model can potentially undergo are important to the detection and recognition of the object in the scene. Hence, the success of the façade detection algorithm is dependent on its ability to identify façade features that are invariant under the perspective transform.

3.2 Invariant Façade Properties

Although buildings can consist of all shapes and sizes, they are predominantly constructed using combinations of flat surfaces that meet at right angles. In particular, in urban environments building surfaces generally have surface markings such as windows, doors, buttresses, color changes, texture changes, and trim that are aligned horizontally or vertically along the building's faces (see Figure 7).

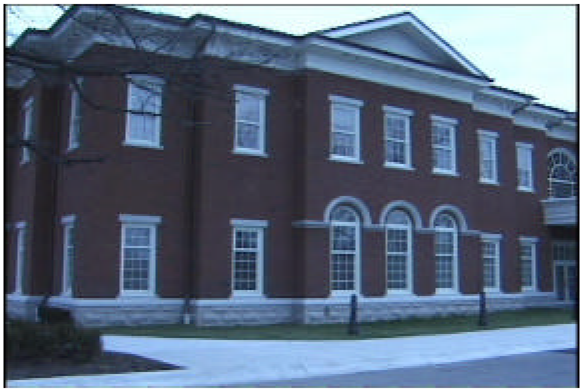


Figure 7. Building demonstrating parallel facade structure in the form of windows, trim, and face boundaries.

These building characteristics, common to many human-made structures, are a key component of our algorithm that will automatically extract these features and align them to the base model. Such regularity can be exploited to extract building facades from imagery. In particular, common properties exhibited by all of these facade features lead directly to the identification of facades in the image by extracting image structure that conforms to these properties.

Building facades consist of straight lines. However, this characteristic is also shared by many objects that are not related to buildings. Although extracting straight lines in a typical ground level scene will most likely segment building boundaries as well as building surface markings, it will also potentially segment straight line features arising from structures such as sidewalks, road edges, and fences. All noise that is extracted with the building facades could decrease the ability of the algorithm to automatically align an image with the model.

The invariant property declaration that buildings are made from straight lines can be refined to say that buildings are made from straight, parallel lines. In most instances, the building characteristics mentioned above consist of lines that are not only straight, but are also approximately parallel in world space. Because a given building face typically has multiple rectangular structures such as windows and doors, as well as elongated rectangular surface markings arising from trim, brick, or other decorative texturing, it can be reasoned that most building faces will consist of relatively dense parallel structure. Although it is true that there are other world objects that exhibit parallel structure, few have as a high a concentration of parallel lines as building facades in our domain. In addition, because we expect that the goal of the user is to capture facade structure in the image, we assume that most world parallel structure in the scene will arise from building facades. Therefore, by finding parallel lines in an image, it is possible to extract buildings without picking up too much unrelated noise.

3.3 The perspective transform

By assuming that building facades can be meaningfully separated from other image structures through analysis of parallel and planar surface structure, the problem of extracting buildings from an image is reduced to the problem of extracting significant parallel lines in the world. However, due to the perspective transform, when an image is taken of parallel lines, the lines are typically not parallel in the image. Rather, the lines converge to a common point referred to as a vanishing point. The only time that parallel world lines will be exactly parallel in an image is when the surface normal of the plane that contains the lines is parallel to the camera normal. The vast majority of the time, parallel lines will converge to a vanishing point. The location of the vanishing point is dependent on the orientation of the lines and on the orientation of the surface that contains the lines with respect to the camera normal. The greater the angle between the camera normal and the surface normal, the more rapidly the lines will converge.

Equation 1 and Equation 2 define this relationship between the camera normal and the normal of the surface containing parallel lines and demonstrate that the only situation in image space parallel lines arise from world space parallel lines is when these normals are parallel. The relationship between the location of a vanishing point $(u,v)^T$, as defined in Equation 1, and line *L*, where $(b_1, b_2, b_3)^T$ are the direction cosines of *L*, is given by Equation 2.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f\dfrac{b_1}{b_3} \\ f\dfrac{b_2}{b_3} \end{pmatrix} \qquad \text{Equation 1}$$

$$L = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + l \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right\} \qquad \text{Equation 2}$$

Note that for b_3=0, indicating that *L* is perpendicular to the camera normal, the location of the vanishing point is undefined. Also, note that as b_1 and b_2 approach zero, indicating a decrease in the angle between *L* and the camera normal, the location of the vanishing point approaches the image center.

Equation 2 points out one of the fundamental difficulties in finding vanishing points in an image. As stated above, when the surface normal of the world plane containing parallel lines is nearly parallel to the camera normal, the location of the vanishing point approaches infinity. Hence, it becomes difficult, if not impossible, to state with any certainty where the vanishing point resides. All vanishing point detection algorithms must deal with this problem. Our approach to handling this situation is detailed in Section 4.2.2. As our algorithm is not dependent on calculating the exact location of the vanishing point, but rather only an approximate location, this problem does not significantly impact the outcome of the approach. Vanishing points are used solely for line grouping. At no point do we attempt to pinpoint the exact location of the vanishing point, in spite of the fact that we can approximate it.

3.4 Challenges of the vanishing point approach

By finding vanishing points in an image and the lines that induce them, it is possible to identify world parallel lines, and in turn, building facades. Unfortunately, any arbitrary pair of image lines may give rise to an intersection that is not necessarily a vanishing point. We refer to lines that are not parallel in the world and incidentally intersect as *incoherent line*s. Hence, before committing to the vanishing point approach, the types of image structure that lead to incoherent lines should be studied.

A key aspect of our approach is its ability to separate arbitrary incoherent lines from the lines that are likely to have arisen from parallel world structure. The incoherent lines arise from three situations. The first of these situations is image structure that contains line intersections due to non-building objects in the world. For example, a Ferris wheel consists of straight lines that converge to a single point. The Ferris wheel lines are not, in reality, parallel. Nevertheless, they would be considered parallel by this approach. However, given a vanishing point we can approximated the relative orientation of the camera optic axis and the planar facade that presumable gave arise to the intersection. We can thus test whether a given vanishing point is likely to have arisen from model structure. The probability of a false vanishing point lying near an expected vanishing point is dramatic and unlikely.

Second, some straight line image features may accidentally pass through a vanishing point or may chance to converge with other random lines in the same location, thus inducing a vanishing point. The probability of this effect happening in sufficient concentration to confuse the algorithm is relatively low and can be further reduced using by the same exclusion technique mentioned above.

Third, and perhaps most important, there are other world objects that contain parallel lines. Examples include fences, light posts, sidewalks, and other manmade features. If these features are located in high concentration in an image, and have orientation similar to that of the modeled structure, then they will likely be extracted as buildings by this algorithm. The risk of these incoherent lines, though real, is softened by the fact that the objects that cause them typically do not overlap building surfaces in images in high concentration. For example, sidewalks, which are located on the ground plane, will not overlap building faces in an image of the building. Utility poles are normally far apart, reducing their effect on the occluded facades. Fences, on the other hand, if located in front of a building in an image, could prevent the algorithm from aligning the image correctly with the model. It is hoped that all of these issues can either be avoided by a trained operator, or that they will not be significant enough to render our automatic image to alignment approach infeasible for a given scene. As shown by Lowe [49][50], there are a set of image features including parallelism that remain invariant over a large set of image views. These features are, of themselves, insufficient information to establish structure in an image. However, they provide evidence of structure and can be used to trigger further processing or to test additional criteria in order to establish the existence of structure. For example, an instance of three lines sharing a vanishing point provides some evidence of structure that is parallel in the world. The existence of additional lines that also share the vanishing point can provide sufficient probabilistic evidence of such structure. In our technique, we assume that facades will be exhibit a high degree of parallel and planar structure, which will produce vanishing points with sufficient evidence to conclude the existence of the facades themselves. We sort the intersections based on how many lines gave support to them and keep only those intersections with four or more such lines.

Chapter 4
Façade Detection

The algorithm used for facade extraction is based on the invariant property that world parallel lines converge to a vanishing point under the perspective transform. This chapter describes in detail the steps of this algorithm.

4.1 Line Segment Extraction

The process of automatic image to model alignment begins with extracting lines from image data. Although facades are primarily composed of horizontal and vertical features, under the perspective transform, the lines associated with these features can be transformed to all possible orientations. Therefore, in order to extract all facade lines, the line finder must be capable of extracting lines of arbitrary orientation. A line finder that is well suited to this task is the orthogonal regression line finder [51][52]. This line finder is not only capable of extracting lines of arbitrary orientation, but also of arbitrary length. The implementation of the orthogonal regression line finder used for the results of this thesis comes from the Horatio image processing package developed by McLauchlan [53]. A detailed discussion of orthogonal regression is beyond the scope of this paper. The basic idea is based on perceptual grouping techniques (see Section 4.2.1.1). The line finder is run with a low RMS threshold for our approach due to the need to extract lines with as great orientation accuracy as possible. In addition, segments less than 20 pixels in length are discarded.

4.2 Line Pencil Extraction

After lines are extracted from an image, the lines are filtered to extract groups of lines that share a common vanishing point. These groups, called line pencils, are considered likely to have arisen from common parallel structure in the world. If it is known that the image being filtered contains a facade, then it is likely that the line pencils produced by the filter arose from the facade. The vanishing point filtering algorithm consists of four main steps. These steps are first, grouping collinear line segments into single lines; second, searching for vanishing points; third, grouping line segments by vanishing point; fourth, hollowing out vanishing point groups.

4.2.1 Collinear Line Segment Combining

When lines are extracted from an image of a real facade, it is common for long lines to be extracted as disconnected collinear segments. This is caused by variations in surface texture, lighting, obstructions, and other "noise" effects [54]. Each of these extracted line segments will be nearly, but not exactly collinear. The shorter the segment, the greater the potential magnitude of orientation and endpoint location errors. During vanishing point searching, the line segments will be extended beyond image boundaries to form intersections with other lines. Even small errors in line segment orientation can lead to large errors in intersection location. Therefore, such collinear segments should be combined into one line in order to reduce not only the overall number of lines, but also to minimize small segment error magnification.

4.2.1.1 Line Segment Collinearity Constraints

Line segments are grouped based on orientation and distance constraints (see Figure 8). This grouping process is similar to that of Boldt [51] and other perceptual organization techniques [49][50] that group image features based on image features that tend to be invariant over a large set of views, such as collinearity and parallelism. In order for two segments to be considered collinear, both constraints must be satisfied. The orientation constraint is a maximum allowed difference in angular orientation. The slope of line segments is converted to degrees in order to make this difference calculation. The distance constraint is a maximum distance from the endpoints of one line segment to the line containing another segment. This value is calculated for two segments by calculating the perpendicular distance from the endpoints of one segment to the line containing the other segment. Note that this definition of proximity is not symmetric. Hence, it is possible for segment A to satisfy the collinearity constraints with a line segment B, while segment B does not satisfy the collinearity constraints with segment A. The orientation and distance threshold values are based on the expected error in the locations of line segments in the image.

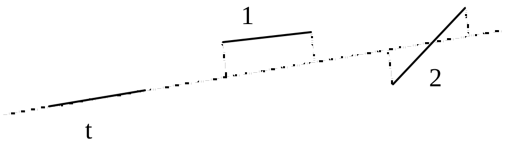


Figure 8. Line segments 1 and 2 are considered collinear with segment t if the distance of the endpoints of 1 and 2 to the line containing t are within the collinearity distance threshold and if the angles between t and 1 and t and 2 are within the angular difference threshold. In this example, although 1 and 2 meet the distance threshold, 2 does not meet the angular difference threshold. Thus, segment 1 would be considered collinear with segment t, while segment 2 would not.

After collinear grouping is performed for every line segment in the image, it is possible for a given segment to belong to multiple groups. Each line segment is then assigned to a single group based on the number of segments in the group. The group with the most segments is considered the “strongest” group, and “claims” all of its contributing line segments. A group “claims” its line segments by removing its line segments from other groups to which they belong. After a group claims its segments, the next strongest group then claims its segments and so on until all segments belong to only one group. During this process, it is possible for some groups to completely disappear as all of their segments can be claimed by stronger groups. The end result is a set of grouped collinear line segments, with each line segment belonging to at most one group.

An example of the grouping result is shown in Figure 9 and Figure 10. The lines extracted from Image A are shown in Figure 9. A representative set of seven approximately collinear line segment groupings is shown in Figure 10. In this particular set, each cluster of collinear segments is composed of up to 37 segments.

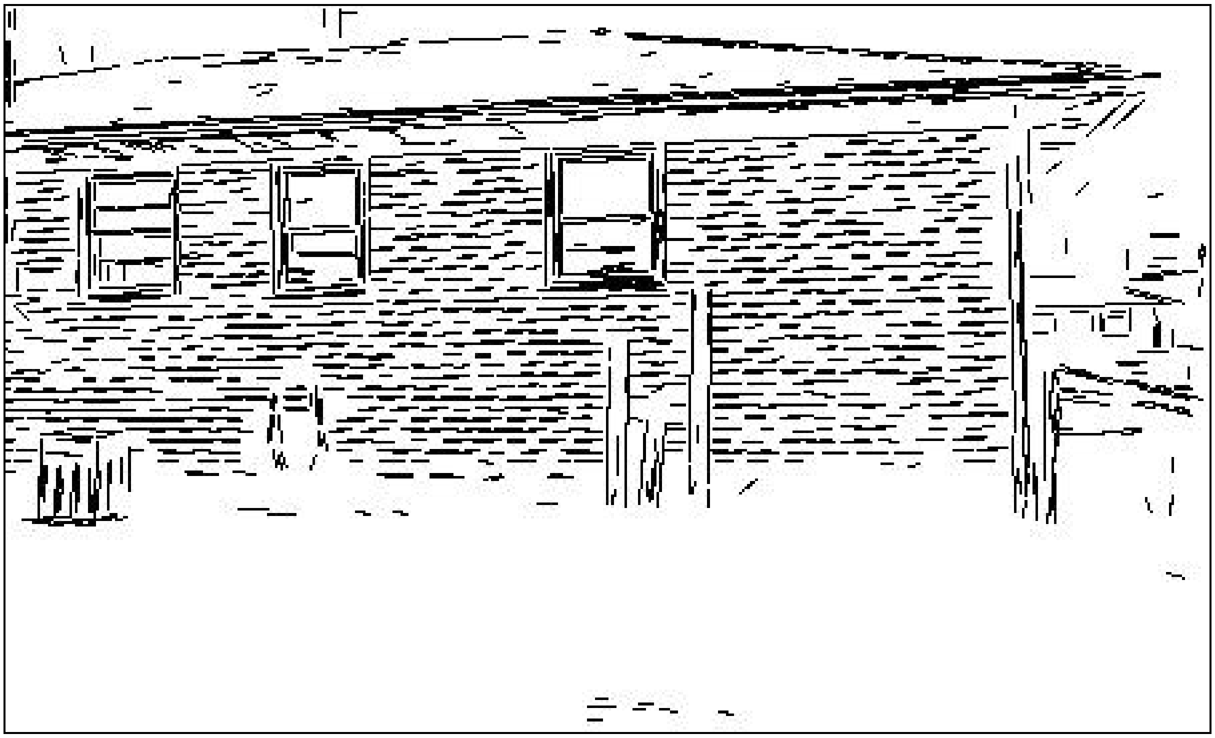


Figure 9. Line segments extracted from Image A by the line finder.

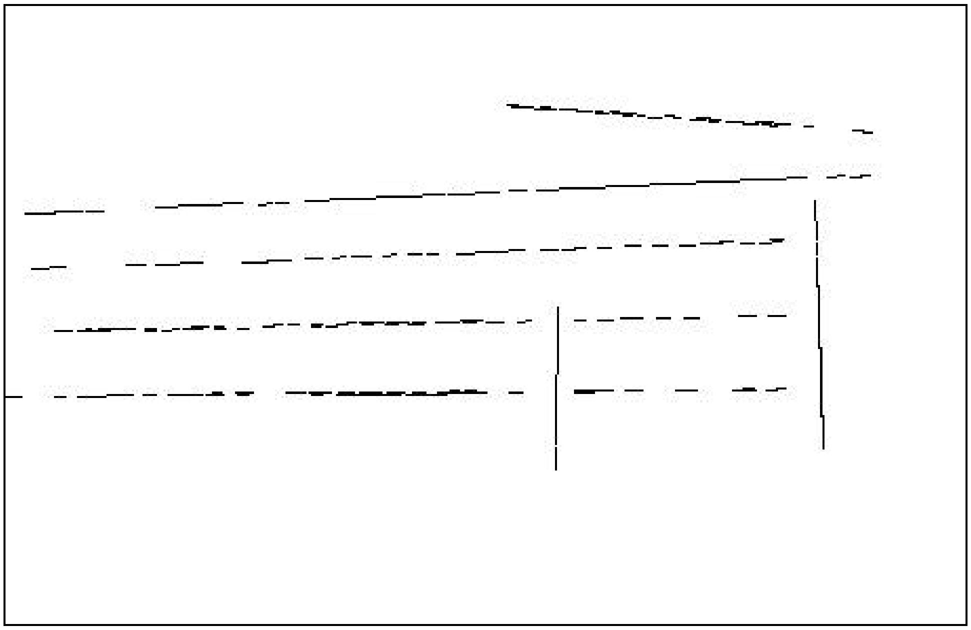


Figure 10. Representative set of approximately collinear segment groups from Image A. In all, over 200 groups were found.

4.2.1.2 Line Segment Averaging

Next, the line segments within each group are averaged to form a single line. The line segments are averaged by converting each segment to (r, θ) coordinate space, by averaging the (r, θ) values, and then converting the average value back to Euclidean space. This averaging method, though usable, does have at least one undesirable characteristic. It is possible for the resulting average line to poorly fit its contributing line segments. This can occur when all of the line segments are oriented in a stair-stepped fashion as depicted in Figure 11. Because most of the line segments have a common orientation, the average orientation of the lines will be close to the orientation of this dominant subset of line segments. This will result in an average line that does not closely fit all of the contributing line segments. The desired average line is the line that passes most closely near all segment endpoints. This goal could be achieved, at the expense of run time, by replacing line segment averaging with a line fitting technique such as least squares minimization.

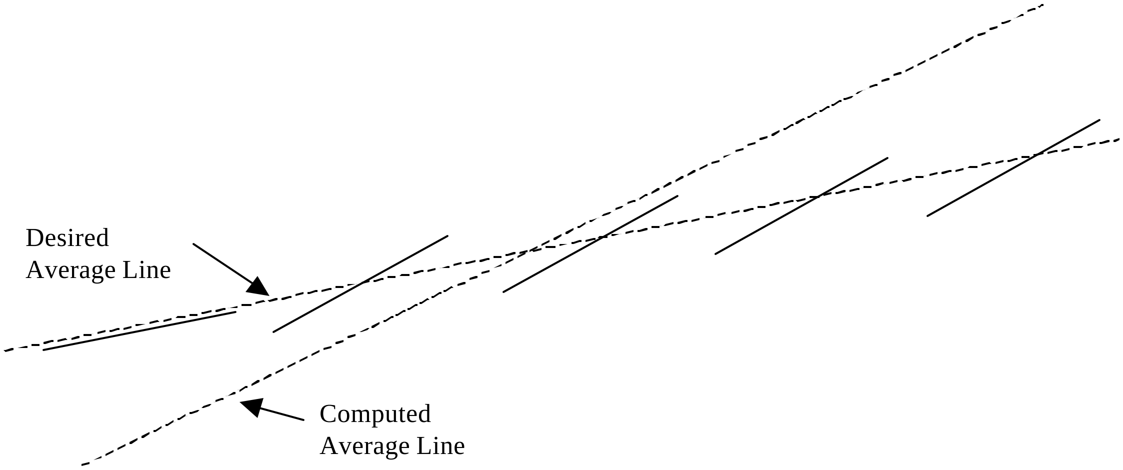


Figure 11. The stair-stepped orientation of the line segments averages to form a line that does not pass near all of the line segments.

Figure 12 shows a representative set of collinear segment groups from Image A and their associated average lines. The segment groups are drawn in black and the average lines are drawn in orange. Note that a few of the line segments do not lie exactly on the average line.

This behavior is expected due to the nature of the collinearity constraints. Segments need only be approximately collinear as defined by the constraints to be grouped together.

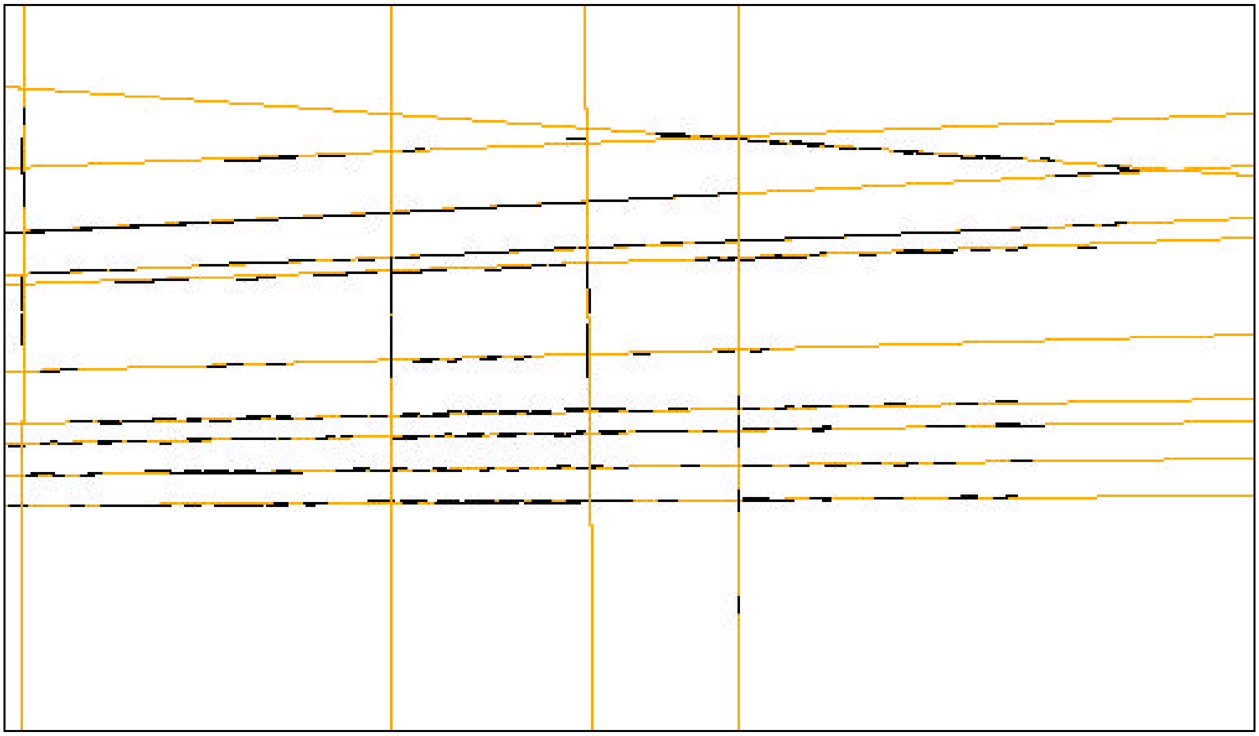
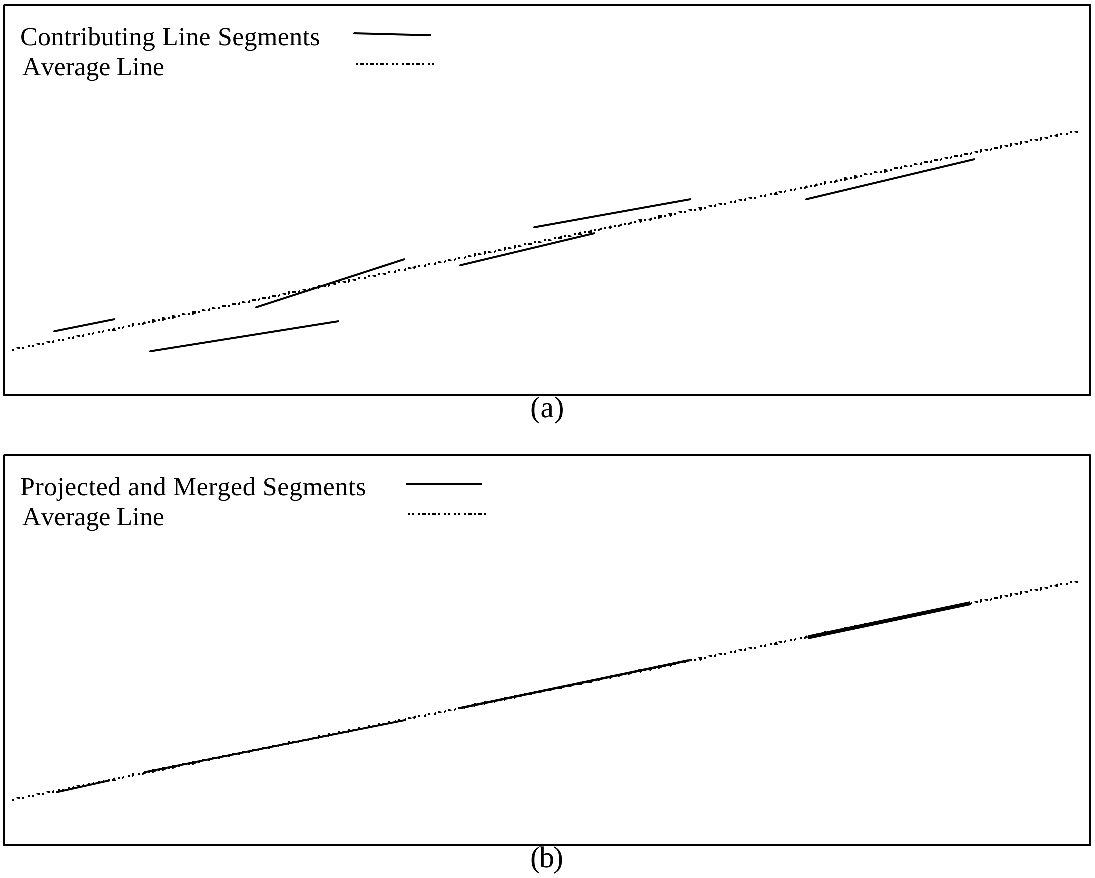


Figure 12. Segment groups (shown in black) and associated average lines (shown in orange) from image A.

4.2.1.3 Strength Threshold

The new lines created by line segment group averaging are screened for a threshold strength (see Figure 13). Lines that are not sufficiently strong are discarded. The strength of a line is determined by perpendicularly projecting the line's contributing line segments onto the line and calculating the percentage of the line that is covered by the projected segments. The threshold itself is somewhat arbitrary. It should be large enough to discard lines that are formed by chance groupings of segments and small enough to preserve groups that are formed from actual collinear segments. In corner regions, where average lines can become relatively short due to clipping by image boundaries, the average lines are considered to have a length longer than their actual clipped length. Without this compensation, nearly all lines in corner regions will be preserved, thwarting the goals of the line segment grouping step.



(a)

(b)

Figure 13. (a) A group of segments and the resulting average line. (b) Segments projected onto the average line and merged. If the average line is not covered by a threshold percentage by its line segments, then the average line is discarded.

Figure 14 shows the lines resulting from running the line segment grouping and averaging algorithm on Image A prior to filtering lines using the strength threshold. Figure 15 shows the line set after applying the strength threshold. Note that only the lines arising from significant straight structure in the image remain.

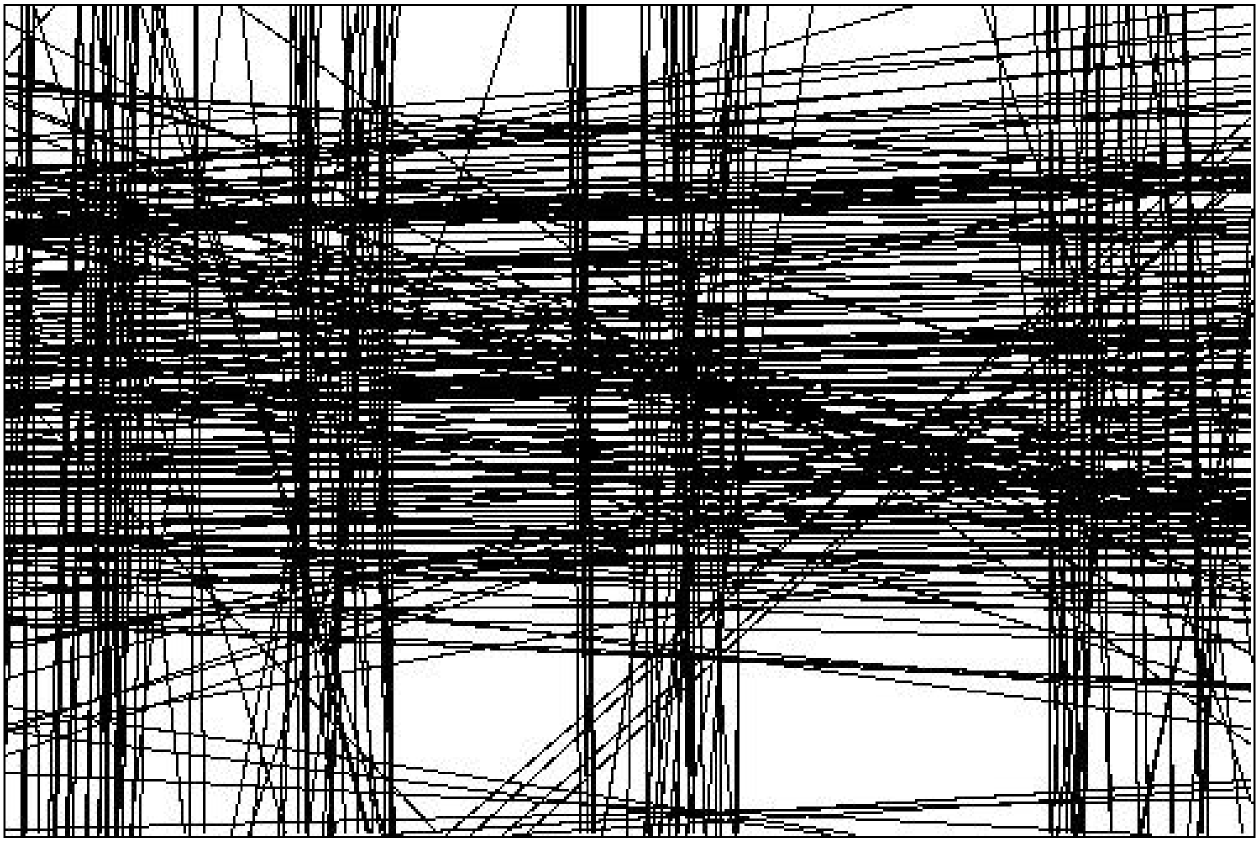


Figure 14. Average lines from Image A prior to applying the strength threshold.

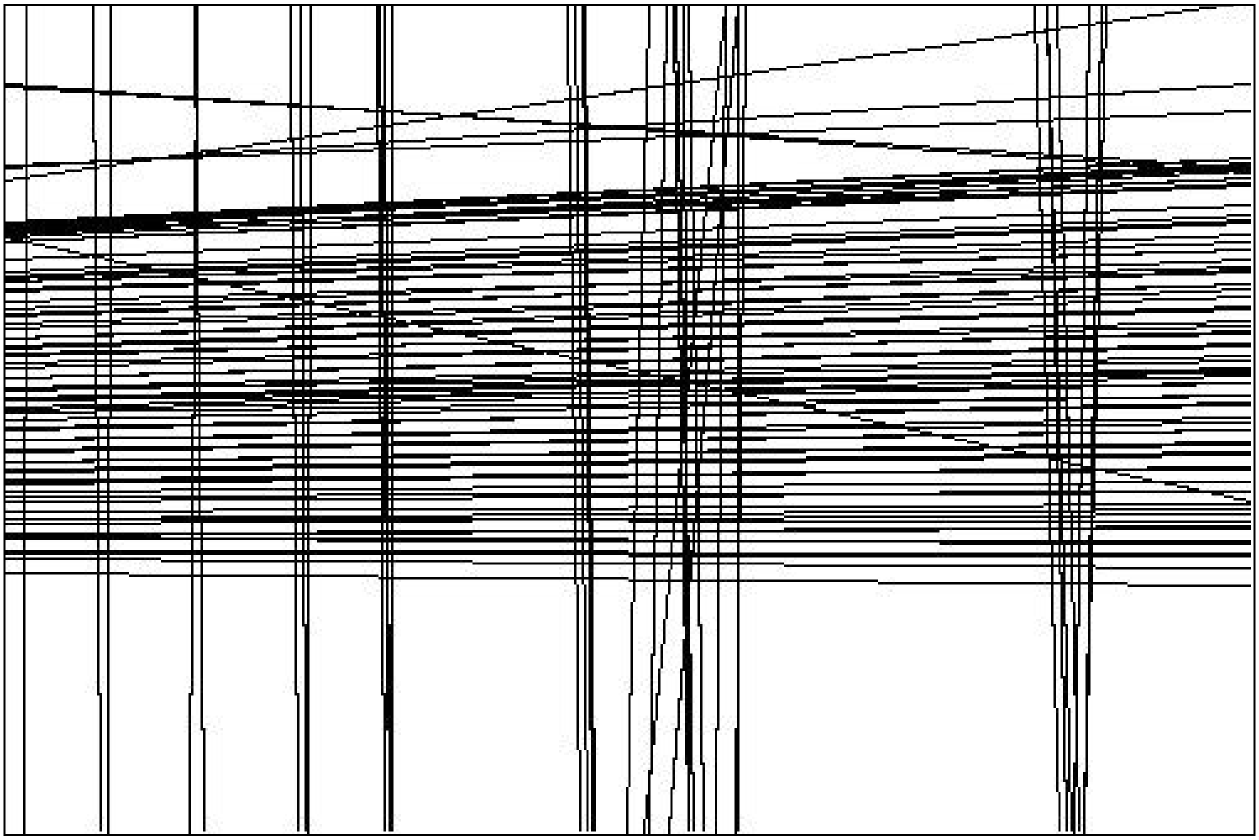


Figure 15. Average lines from Image A that remain after applying the strength threshold.

4.2.2 Vanishing Point Detection

The lines produced by collinear line segment combining are used to identify vanishing point pencils. The basic idea of the pencil identification algorithm is to compute the intersection of every pair of lines and then to group the intersections by proximity. However, this simple concept is complicated by the fact that all line segments contain error. Within the body of a line segment, this error is bounded by the inherent error of the image, the accuracy of the line finder, and the error introduced by collinear line segment grouping. However, beyond the endpoints of the line segment, the error grows linearly with distance from the segment endpoints. Thus, as a line segment is extended far from the image boundaries, the location of the segment becomes less and less certain, making the region of space potentially inhabited by the segment larger and larger. When two such line segments are intersected, the intersection location is affected by the error of the line segments. As the intersection of two line segments moves farther from the image boundaries, it can become impossible to state the location of the intersection with certainty. Rather, it can only be stated that the intersection lies within some region of space that is bounded by the error of the line segments.

4.2.2.1 Error Model

In order to quantify and compensate for the error in lines, it is necessary to develop an error model for lines and for intersections. The error in a line segment can be parameterized as error in the line segment's endpoint locations (see Figure 16).

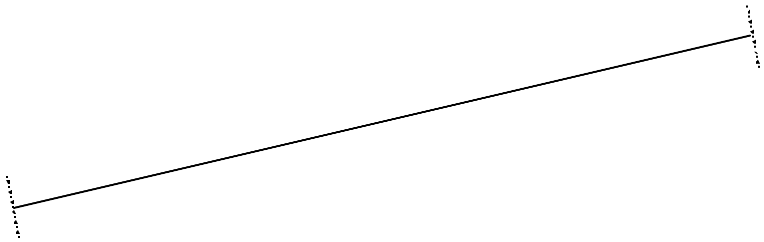


Figure 16. The error in a line segment is modeled as error in endpoint locations. The line segment's actual endpoints could lie anywhere on the dotted lines. The length of the dotted lines is related to image error properties.

The region of space inhabitable by the line segment is bounded by the box formed by perturbing the segment’s endpoints perpendicularly by the maximum expected endpoint error. The region of space inhabitable by an extension of the line segment is bounded by the diagonals of the line segment's error bounding box (see Figure 17).

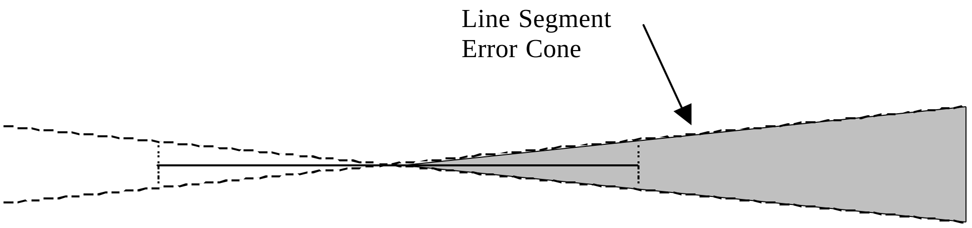


Figure 17. The shaded region represents the region of space inhabitable by the line when accounting for line segment endpoint error. This region is termed an error cone.

This error region is the shape of two symmetric triangles that touch at the center of the line segment. One half of this region, or the region extending from the end of either endpoint, is termed the *error cone* of the line segment. The location of the intersection of two line segments is bounded by the intersection of the line segments' error cones (see Figure 18). This region is termed the intersection's *error box*.

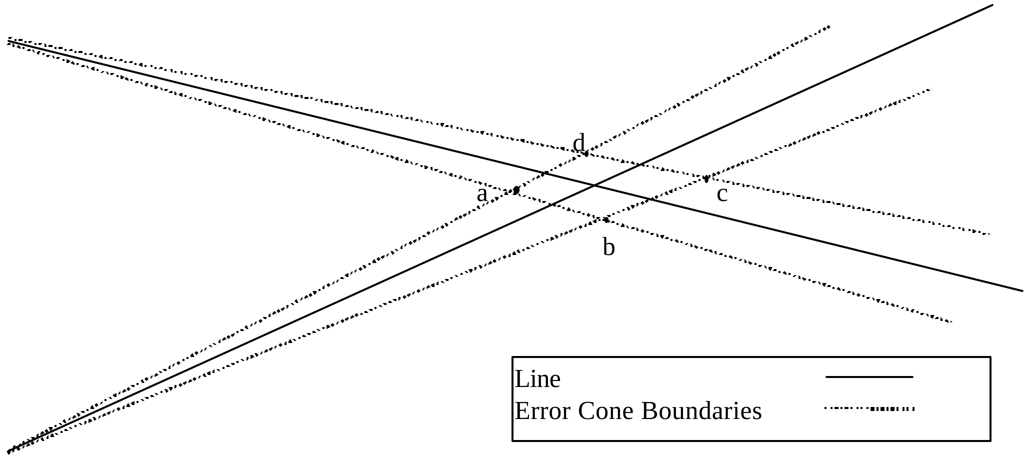


Figure 18. The intersections of the lines’ error cones form the error box. Points a, b, c, and d are the error box vertices.

4.2.2.2 Unbounded Error

The error box intersection error model defines the region of space inhabitable by an intersection. However, this error box can, in some cases, be unbounded. Consider the case where the angular difference between two line segments is less than the sum of half the angular widths of the line segments' error cones (see Figure 19). Note that there is no intersection between the inner error cone boundaries of the lines. In order for the error box to be bounded, the error cone lines of one line must both intersect both lines of another line's error cone.

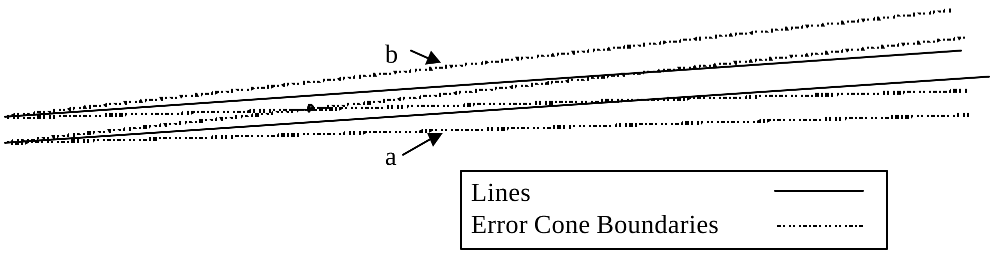


Figure 19. Error cone lines a and b do not intersect in the direction of the lines' intersection. This results in an error box of infinite size.

When an error box is unbounded, there is no longer any certainty in the intersection's location. When intersections are being grouped together in the next step of the algorithm, this uncertainty in location can lead to problems. On one hand, the intersection may be so far from any other intersections that it will not be grouped at all. On the other hand, so many intersections may fall within the intersection's error box that many intersections could be falsely grouped together. In order to avoid these uncertain conditions, intersections with unbounded error boxes are either discarded or given the special designation of "parallel intersection".

If all unbounded intersections were simply discarded, then no image space parallel line groups would be extracted by the vanishing point filter. If a group of lines are parallel in an image, then the intersection error box of any two lines in the group will be unbounded. Hence, simply discarding these intersections will effectively discard all image space parallel lines from vanishing point extraction, and, in turn, from the results of the façade extraction algorithm. In order to preserve the intersections of image space parallel lines, these intersections are added to their own intersection pool. Rather than calculating the exact location of these intersections, they are assigned an orientation value based on the orientations of the contributing lines. During the vanishing point grouping step, the parallel intersections are grouped based on orientation only, rather than on physical proximity.

4.2.2.3 Intersection Grouping

After all intersection error boxes are formed, the intersections are grouped based on proximity. Rather than attempting to segment the infinite intersection space into regions, the intersections are grouped by cohabitation of error boxes. Two intersections are grouped together if the center of each intersection's error box lies within the other intersection's error box (see Figure 20). Note that the center of an error box is defined as the intersection of the line segments disregarding error, i.e. the intersection of the error cones' axie. This definition is chosen as a consequence of the shape of error boxes. Error boxes of nearly parallel lines are long and narrow. These error boxes, by nature of their length, can often contain completely unrelated intersections (see Figure 21). If intersections are grouped by overlap of the error boxes alone, then these long error boxes will engulf lots of intersections and have the highest intersection group strengths. The end result would be groups of lines that do not converge to a single point, but rather to a long region of space.

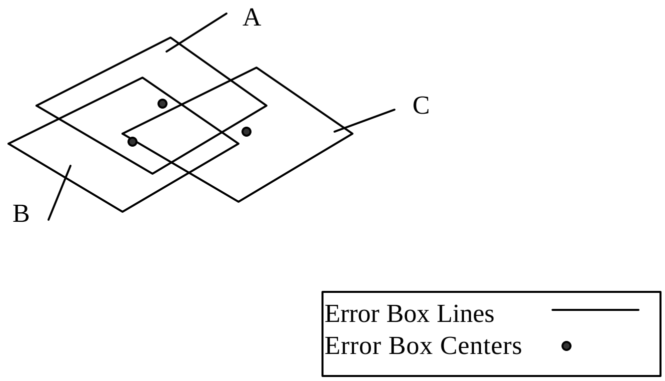


Figure 20. The centers of error boxes A and B both lie within each others error boxes. These intersections are grouped together. However, neither center lies within error box C, excluding C from the intersection group.

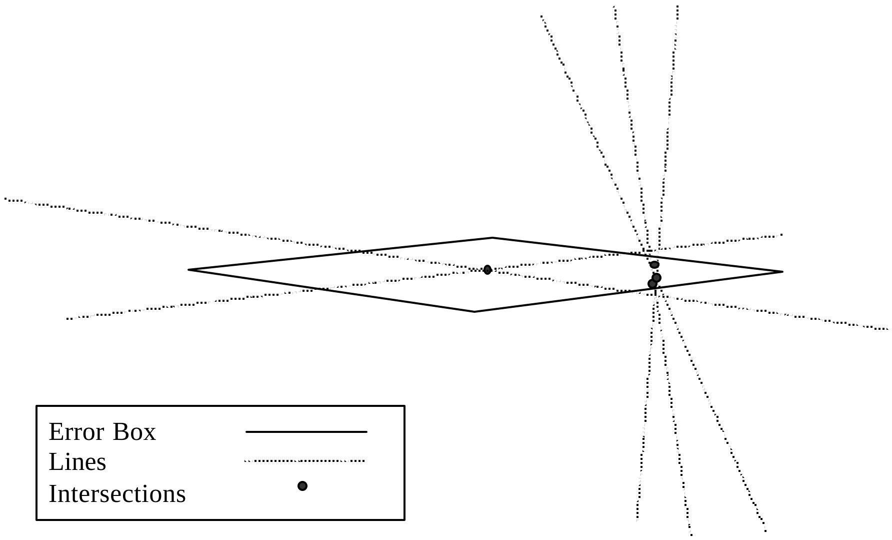


Figure 21. Although the error box contains the group of intersections at the right, the error box lines do not correspond to the same vanishing point. Thus, intersections are grouped only by mutual cohabitation of error box centers in the error boxes.

Once all possible intersection groups are formed, the intersection group with the highest number of intersections "claims" the line segments that form its intersections. The claimed line segments are then removed from weaker groups that also claim them. This claiming process continues, with the strongest remaining group getting the next claim, until all groups have claimed or have been deleted by virtue of other groups' claims. The end result is groups of line segments with the segments in each group passing through a common region of space, i.e. a vanishing point line pencil.

Figure 22 shows the result of vanishing point extraction and grouping on Image A. Each line pencil is displayed in a separate image. Although 7 line pencils were extracted from Image A, only 4 are shown because three of the groups consisted of less than four lines each. Groups having only a few lines can be discarded because they do not contain sufficient evidence to conclude from our assumptions that they arise from world parallel structure (see Section 3.2). In Figure 22, group (a) arises from facade structure that is parallel to the ground plane in world space. Group (b) arises from structure that is perpendicular to the ground plane in world space.

Group (c) also arose from structure that is perpendicular to the ground plane, but had orientation sufficiently different from group (b) to not be grouped together. Group (d) is an intersection cluster arising from the intersection of roof planes. This group can be eliminated based on the expected location of vanishing points due to the relative orientation of the model faces to the camera optic axis (see Section 3.4)

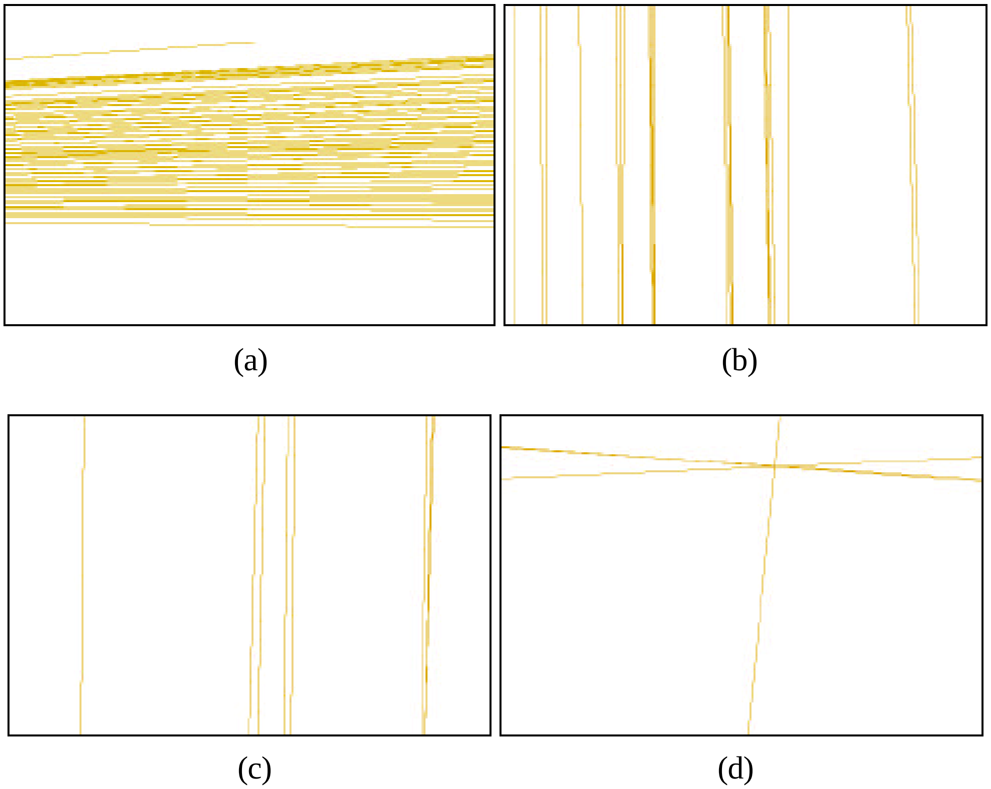

(a) (b) (c) (d)

Figure 22. Line pencils extracted from Image A sorted by the number of lines in the pencil. Group (a) had the most lines while group (d) had the fewest lines.

4.2.3 Vanishing Point Group Thinning

Some surface textures, such as brick, have an abundance of parallel surface features. The resulting vanishing point group may contain many lines that are spatially separated by only a short distance. It is likely that most of these surface features will not be represented in the simple models used as input to this algorithm. Because the starting location of the alignment minimization is only an approximation, it is possible that the model edges could incorrectly align to these interior surface features. Therefore it is desirable to decrease the number of these features contained in the vanishing point filter output.

The number of lines in a vanishing line pencil can be decreased using the same technique as was used to eliminate vanishing points based on image location (see 3.4). By utilizing the approximate camera parameters provided as input to the algorithm, we can predict the location of projected model edges in image space. We can thus eliminate lines that are not in regions that are likely to be inhabited by a projected model line. However, it is still possible for a dense set of lines to be located within these regions of high probability, which could result in local minima in the alignment objective function (see Chapter 5). To reduce the probability of dense clusters of lines in this situation, we can decrease the density of these lines by using the following heuristic approach.

The heuristic for reducing the number of lines in a vanishing point group is "If there is a dense bundle of lines, then only the boundary lines of the bundle need to be preserved". In other words, given a sorted set of lines in a vanishing point group, eliminate lines that are relatively close to both neighbors in the sorted set. This heuristic is based on two observations. First, even if a dense bundle does contain a line that corresponds to a model line, then each of the other lines will represent a local minimum in the error function that will likely thwart the alignment process. Second, if there is a pattern of dense lines, then it is likely that the lines arose from the same surface. Because the model generally contains only surface boundary lines, then the boundary lines of the group are likely to correspond to model lines.

A line group is thus thinned by sorting the group’s lines relative to the shared vanishing point, searching for bundles in the group, and removing the lines interior to bundles. First, the lines are sorted by orientation relative to the vanishing point. In some cases, such as when two lines intersect, it is not clear what the sorted order of the lines should be (see Figure 23). In order to eliminate these situations each line is treated as if it were formed from the vanishing point error box center and the line endpoint farthest from the vanishing point. This eliminates all intersections and establishes an absolute ordering of the lines. Second, the line group is split into subgroups based on the median angular difference between adjacent lines in the group. Gaps between adjacent lines that are larger than the threshold act as subgroup separators. Third, the lines interior to each subgroup are discarded (see Figure 24). Finally, the subgroups are merged back together to form a single group. The result is the original group with certain regions of the group corresponding to dense bundles removed.

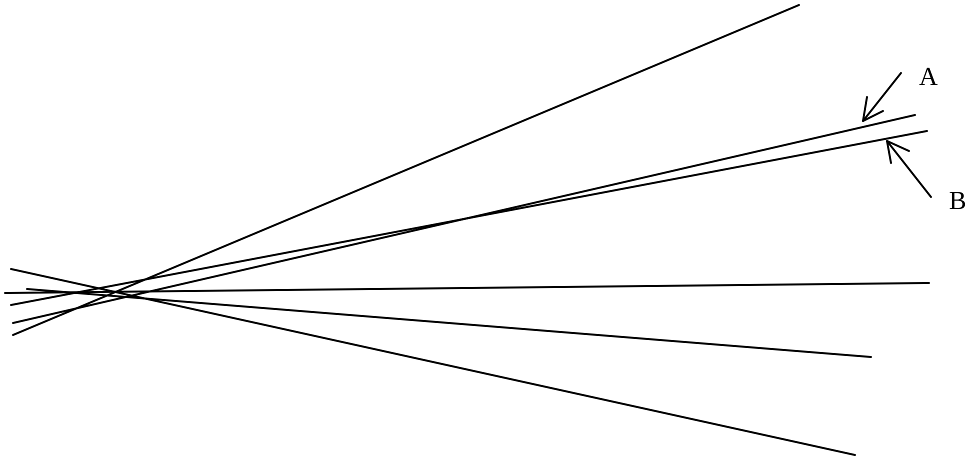


Figure 23. The relative ordering of segments A and B is not clear. New lines are created from the vanishing point and the endpoints of A and B, which removes the ambiguity.

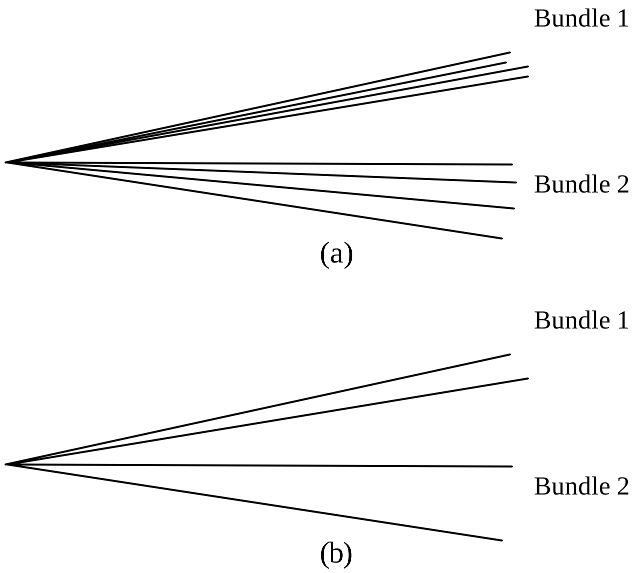
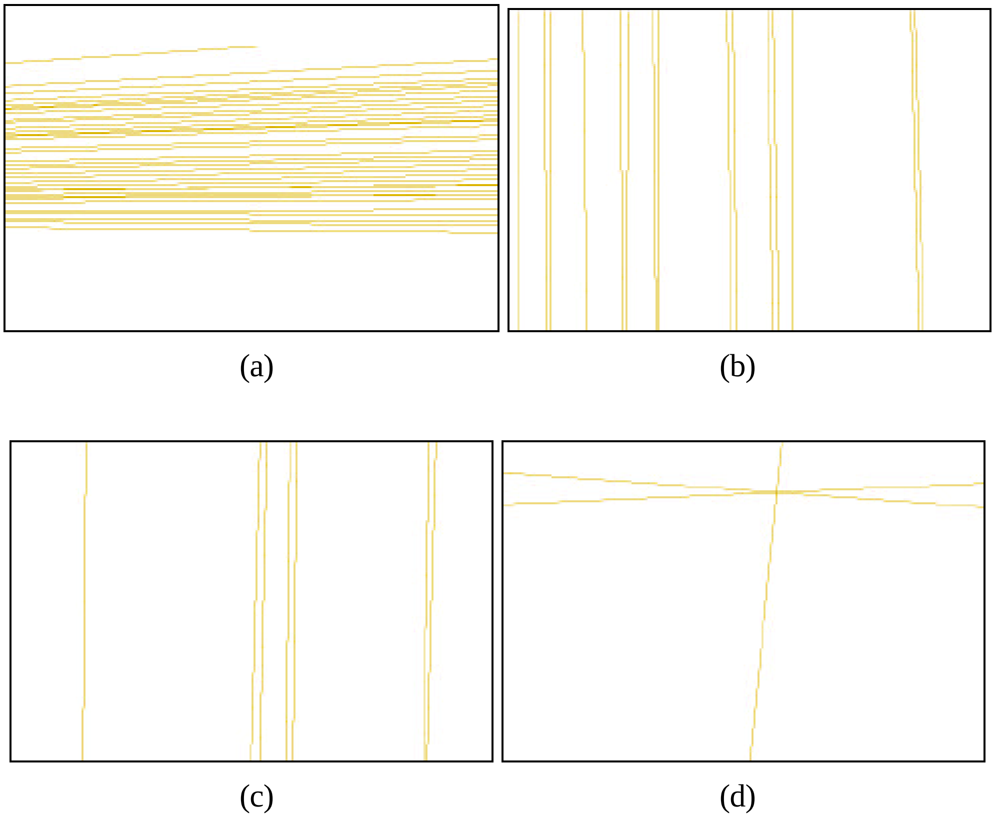


Figure 24. (a) Bundles 1 and 2 will be separated in the first thinning step. Next, bundles 1 and 2 will be hollowed out. (b) The lines remaining after the thinning step is complete.

Figure 25 shows the line pencils extracted from Image A after this thinning step. Note that the densely packed line pencils are affected the greatest by the thinning step. The number of lines in pencil (a) in Figure 22 was reduced from 72 to 35, whereas the number of lines in pencil (c) was reduced from 8 to 7.



(a)

(b)

(c)

(d)

Figure 25. Result of thinning the line pencils extracted from Image A and shown in Figure 15.

Now that we have extracted sets of lines that are likely to have arisen from structure that is represented in the model, we proceed with a discussion of how alignment of these lines with the model is to be performed. We begin with a general discussion of multivariate alignment and then define the objective function we use for the alignment.

Chapter 5
Automatic Alignment

The human vision system has the ability to quickly evaluate images for various qualities. For example, if lines are extracted from a region of an image, it is relatively easy to find the region given only the extracted lines. Likewise, it is relatively easy to hand align an image of a model with the model. On the other hand, accomplishing this task programmatically is difficult due to the complexity of constructing a function that accurately reflects the degree to which an image is aligned with the model. Such a function would be relatively easy to construct if image line to model line correspondences were known. However, these correspondences are not known, requiring the use of a general technique.

Given the set of lines produced by the techniques in Chapter 4, model alignment must compute the external camera parameters that best align the model with these lines. To accomplish this, it must search the camera parameter space consisting of six degrees of freedom, requiring that the alignment routine minimize over six variables. We begin with a general discussion of multivariate alignment and the desirable characteristics of an objective function before proceeding with a discussion of the details of the objective function used by our algorithm.

5.1 Multivariate Minimization

Developing a function for measuring alignment quality requires an understanding of multivariate minimization techniques. Most multivariate minimization techniques operate by measuring the value of the function at a point, perturbing the point, determining the difference between the new function value and the original value, and perturbing the point again based on the function values. Conceptually, the minimization routine searches across the surface of a function looking for the function's minimum value. If there is a local minimum in the function, then the minimization technique may falsely report the local minimum as the overall minimum. If there are discontinuities in the function, then the behavior of the minimization routine can become unstable. Thus, when determining an objective function for aligning images with models, care should be taken to reduce the probability of local minima in the function and of discontinuities in the function. In particular, these characteristics should be avoided near the alignment location. In addition, the minimum value of the fitting function should be located at the alignment location of the image and the model, thus ensuring that the function will correctly recognize the aligned condition. If a function can be developed that satisfies these requirements, then a minimization technique should be able to find the alignment location.

5.2 Objective Function

We now consider an approach to defining an objective function for the purpose of aligning the extracted line set (see Chapter 4) with the base model provided as input to the algorithm (see Section 2.4). In order for alignment to occur, the object function should be minimum at alignment and smoothly increasing from there. Ensuring these requirements are met is a good starting point for defining the function. In the precisely aligned state the model lines should align exactly with image lines. The error of this state should be zero, provided that all model lines are represented in the image. It is assumed that there are no other orientations close to the aligned orientation that will align the model with image lines as accurately as the actual aligned orientation. Thus, in the region near the aligned orientation, the error function value is expected to be greater than zero. In fact, as the orientation of the camera is changed even slightly from the aligned position, model lines and image lines begin to diverge, lending credibility to this assumption. Hence, by basing the value of the objective function on the distance between model lines and image lines, it is possible to create a function that will be zero at the aligned location and non-zero in the region near the aligned location.

In some cases, model lines will not directly correspond to lines in an image. Lines can be excluded from an image by occlusion, lighting, or other noise effects. In these cases, the error function will not be zero at the aligned location. Rather, the value will be related to the distance from the unrepresented model line to the nearest image line. If the unrepresented model line is moved closer to the closest image line while disturbing the remaining model lines only slightly, then it is possible for the error function value to decrease. In other words, when model lines are not represented by image lines, the minimum value of the error function may not be at the aligned location.

By scaling the error function it is possible to decrease the probability that the error function minimum will reside at some location other than the correct aligned location. The basic idea is to give nearly aligned lines a greater effect on the error function value than lines that are not nearly aligned. If several lines are nearly aligned, then the combined weight of several unaligned lines should be required to pull the lines out of alignment. This alignment weighting can be accomplished by translating the distance measurement through a function, such as a logarithm, that changes the rate of increase of the error value. By using this technique the error value can be made to decrease and increase rapidly as lines come into and out of alignment. If most of the model lines are represented by image lines, then the combined "weight" of the aligned lines will outweigh the error contributions of the unrepresented lines. Note that for this technique to be effective, the slope of the error contribution of an unrepresented line must be relatively small compared to the slope of the error contribution of a represented line. A constant factor can be used to ensure that the error function rate of increase decreases rapidly outside of the region that is near alignment.

Based on these arguments, the alignment objective function is defined in Equations 5, 6, and 7. Equation 5 defines the raw error value for a single, visible, model edge endpoint.

$$P(pt, N, C) = \sqrt{\frac{(N \bullet (C - pt))^2}{(N \bullet N)}}$$

Equation 3

The distance from the endpoint of the model edge (*pt*) to an image segment is measured as the perpendicular distance from the endpoint to the plane containing the image segment and passing through the camera center (see Figure 26).

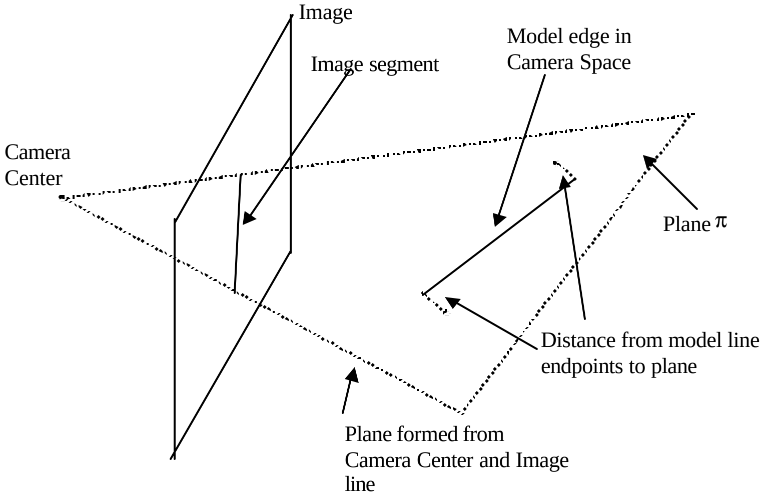


Image
Image segment
Model edge in Camera Space
Camera Center
Plane π
Distance from model line endpoints to plane
Plane formed from Camera Center and Image line

Figure 26. Image line to model line distance is measured from the model line in camera space to the plane formed from the image line and the camera center.

Note that this measurement is performed in camera space, meaning that the model projection onto the 2D image plane need not be performed. The two endpoint distances are then combined as defined in Equation 6.

$$D(M_i, I_j) = \log\left(1.0 + \left|P\left(M_{i_0}, I_j, C\right)\right|\right) + \log\left(1.0 + \left|P\left(M_{i_1}, I_j, C\right)\right|\right)$$

Equation 4

The *log* function is used to ensure that the error value grows rapidly near the aligned location and more slowly as the segment moves out of alignment. As described above, this technique helps ensure that a good alignment is not outweighed by the combined weights of several close but invalid alignments. Finally, the minimum measured error value between a given model line and all image line segments contributes to the final objective function value as shown in Equation 7. The individual error values are combined using a simple sum.

$$E = \sum_{i=0}^{n-1} \min_{j=0}^{m-1} \left(D\left(M_i, I_j\right)\right)$$

Equation 5

Equation 5. Objective function E definition. M_i is visible model line i; I_j is the normal of the plane containing image line j and the camera center C. P(pt, N, C) is the perpendicular distance from point pt to the plane containing C and having normal N.

Note that this error computation occurs in camera space, not in image space. By performing this computation in camera space we avoid errors introduced in the projection from camera to image space [55].

Figure 27 and Figure 28 show the behavior of the objective function for the trivial case of one model edge and one line segment. As model segment *m* moves away from the plane containing image segment *i* and passing through the camera center, the value of *D* increases logarithmically. Figure 29 shows the behavior of the objective function as model line *m* with endpoints initially lying in the plane containing segment *i* rotates out of alignment.

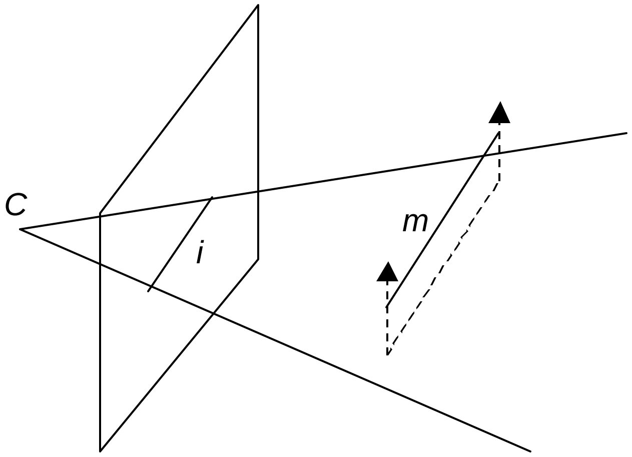


Figure 27. Model line *m* moving away from aligned position with image segment *i*.

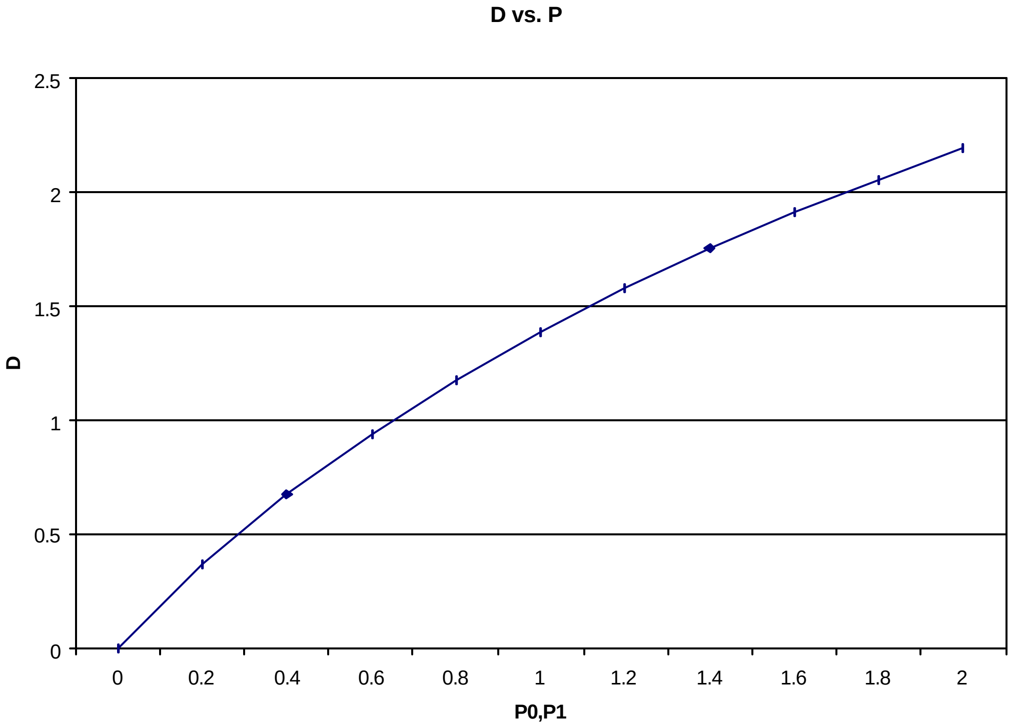


Figure 28. Value of *D* given equal values of P_0 and P_1.

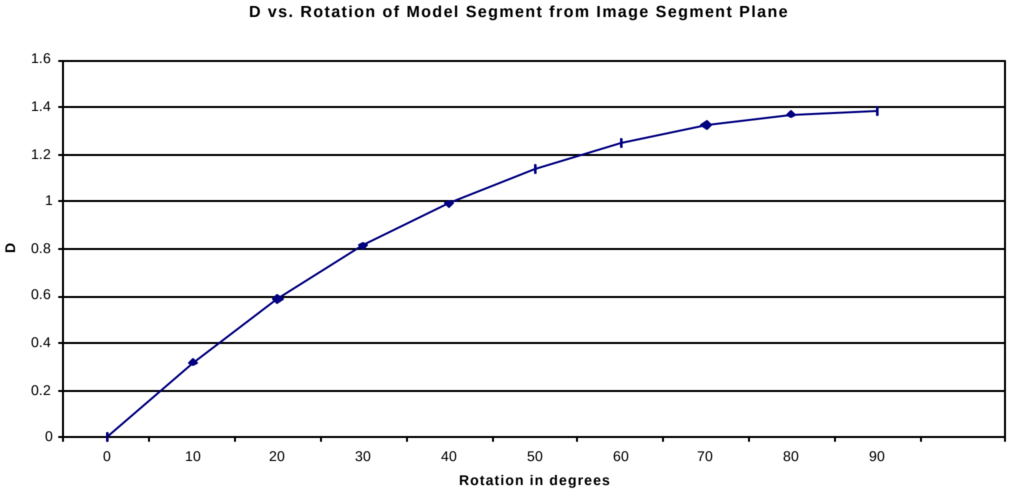


Figure 29. Value of D as *m* rotates away from segment plane *i*.

Figure 30 depicts a slightly more complex, and yet still trivial situation with a single, triangle and an associated projection. The value of *E* as the triangle comes out of alignment with the associated image lines over two of the six external camera parameters is shown in Figure 31. Note that the minimum of *E* lies at the aligned location and that *E* is monotonically decreasing in the region near the aligned location. This indicates that *E* satisfies the requirements for an objective function for this simple dataset.

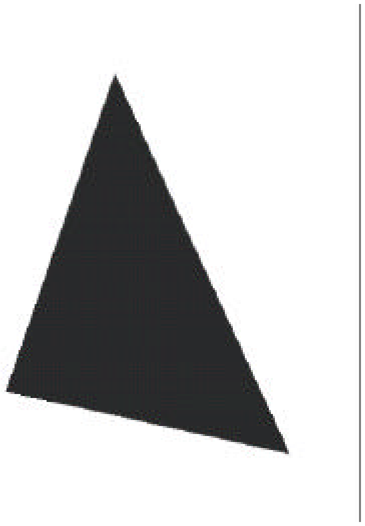


Figure 30. Image of a model consisting of a single triangular patch.

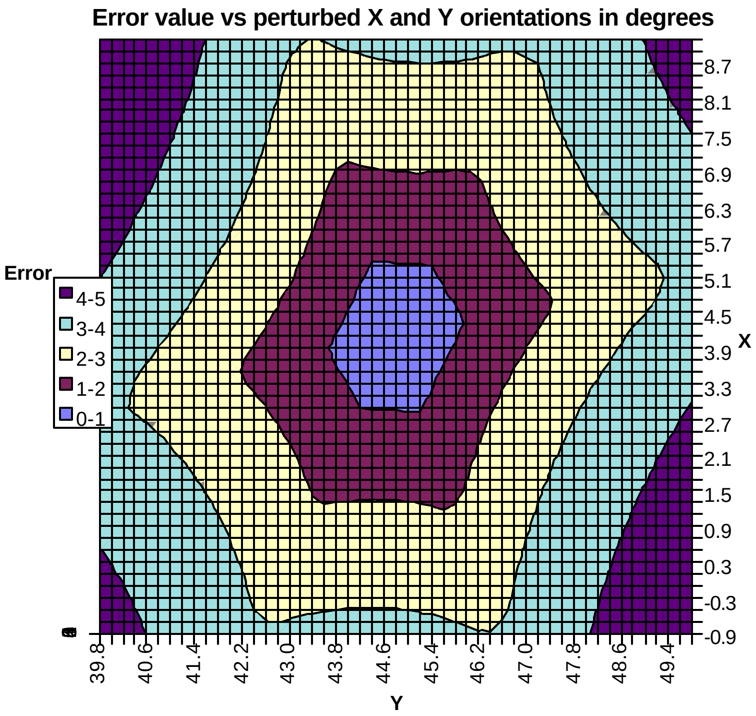


Figure 31. Value of objective function as the X and Y orientation parameters are perturbed about the aligned camera position. The function minimum lies at the center of the graph, which is also the aligned camera position.

Next, consider the situation depicted in Figure 32, in which a model consisting of two faces is viewed such that one of the faces is barely visible. The value of *E* as the face drifts out of view is shown in Figure 33. There is a sharp discontinuity in the function value along the edge of space at which the face is barely visible. This example shows the need to avoid face transition boundaries for isolated, single views.

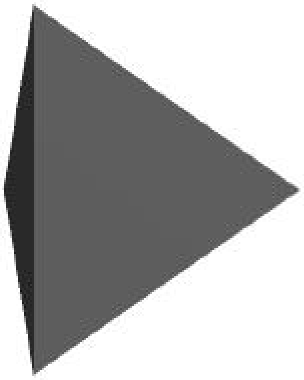


Figure 32. Image of two triangular patches meeting at a right angle. The camera is positioned near the face transition boundary for the left face.

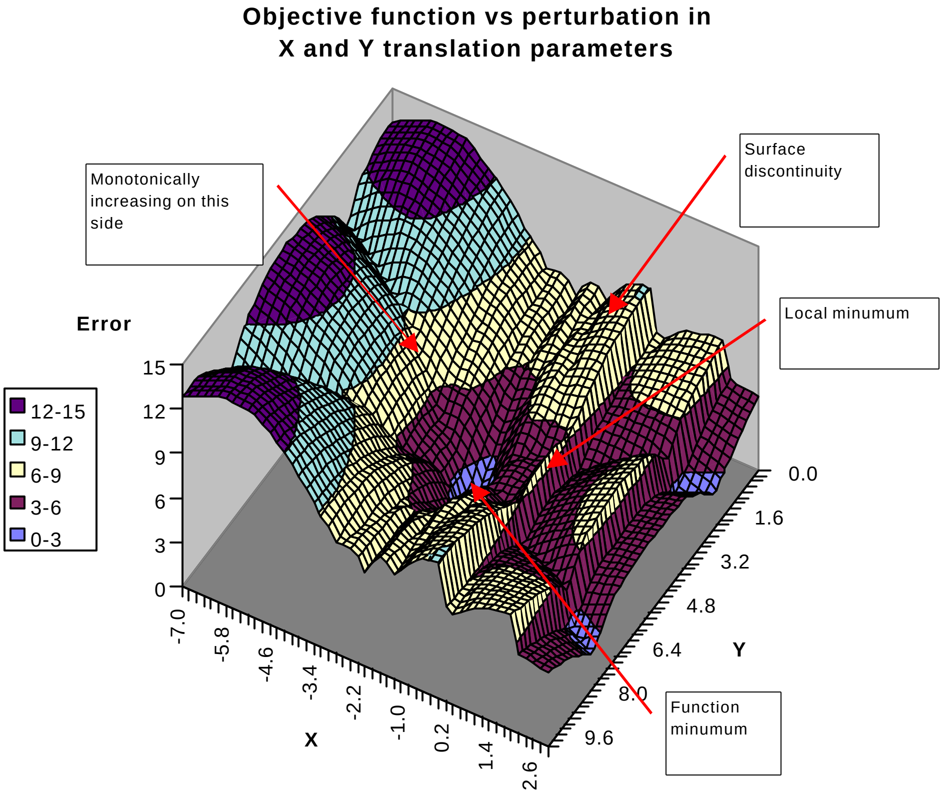


Figure 33. Value of objective function for perturbed X and Y translation parameters for the scene shown in Figure 32. The sharp discontinuity to the right of the center of the graph corresponds to a face transition boundary.

5.3 Objective Function Fitness

In the general case, the defined objective function meets some of the multivariate minimization objective function requirements and falls short on others. In particular, the function's minimum value is typically located at the aligned location and the function is continuous as long as model faces do not appear or disappear, i.e. transition from back faces to front faces and vice-versa. However, the function does not guarantee that there are no local minima near the aligned location. Nor does it guarantee that the minimum value will be at the aligned location and that there are no discontinuities near the aligned location. Thus, there will be situations in which the minimization routine cannot be expected to converge to the aligned location. For example, reconsider the case where a model face is just barely visible in an image. If the starting point of the minimization is on the side of the discontinuity that does not contain the actual aligned position, or even if the starting location is near the boundary, then the iterative minimization process may chose the wrong slope to follow resulting in a misalignment.

5.4 Photo-realistic and Hybrid Rendering

After determining camera parameters that align an image with a model, the image can be projected onto the model from the camera location to give the model a photo-realistic effect. Several techniques have been developed for mapping image textures onto surfaces [11][12][13]. These techniques range from cutting textures against facade boundaries to selecting pixel values from images based on proximity to the image camera location.

Alternatively, a non interpolative technique can be used in instances where a large number of views are available by utilizing the observation that an image projected onto a structure will yield an accurate visualization of the structure within a locus about the camera location. As a viewer enters a scene, the image most nearly matching the current viewing parameters can be loaded and projected onto the base model to yield an accurate representation from the current viewing location. As the viewer moves through the scene, other images are loaded and projected to continue to provide an accurate visualization at new viewing locations

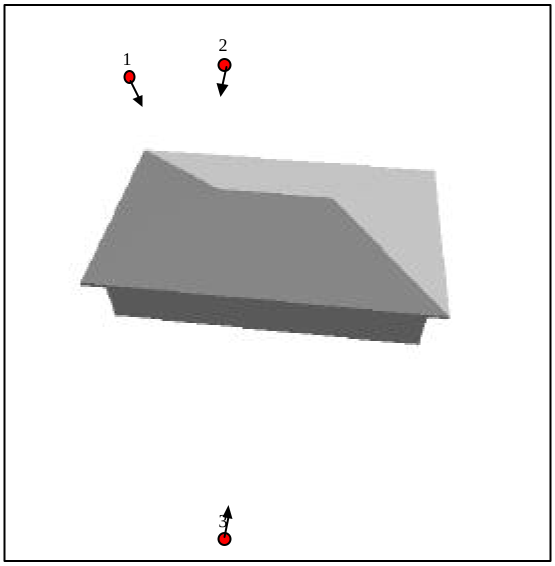
For the examples shown in this thesis, texture mapping was performed using OpenGL [56]. OpenGL integrates projective texture mapping into its functional architecture making it very useful for 3D visualization using complex models and imagery.

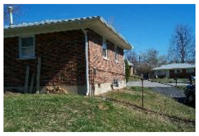
Chapter 6
Results and Analysis

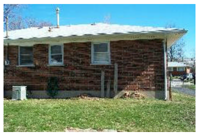
Evaluating the performance of the automatic alignment algorithm requires careful quantitative analysis as well as subjective measures such as visual correctness. It is straightforward to determine the quality of the alignment by visual inspection; however, accurately evaluating the alignment is hampered by the difficulty of developing an alignment objective function. The final fitting error between an image and a model for a particular image-model pair does not indicate whether the error arises from multiple unaligned model edges or a single unaligned edge. Therefore, rather than formulating a single correctness metric, we evaluate each step of the algorithm independently and then attempt to quantify the overall results based on these incremental evaluations.

6.1 Experimental Setup

The algorithm was tested on two datasets referred to as the *house* dataset (see Figure 34) and as the *Hardymon* dataset (see Figure 36). The house scene consists of a simple, residential dwelling and various occluding objects such as shrubs and trees. The residential dwelling is a "ranch" style house with four primary walls and a four sided, sloping roof. Windows, doors, and trim add surface features to the brick veneer exterior. A model of the house was created using hand measurements of primary building features (see Figure 1). The model, which is measured in units of feet, is accurate to within 3 inches at any vertex through physical measurements. Images of the house were acquired using a digital camera set at 1440x960 resolution. Images of the building were taken at the locations indicated in Figure 34.



1 

2 

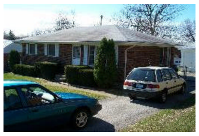
3 

Figure 34. Location of images taken of the house scene. Location 1 is below the level of the ground. Location 2 is about eye level above the ground. Location 3 is a few feet above ground level.

The Hardymon dataset includes a three-dimensional model and images taken of the Hardymon building located at the University of Kentucky. The model was constructed from architectural drawings and is also accurate to within 3 inches at any vertex[1]. The Hardymon model includes only the façade features from the ground up to the bottom of the trim that is located along the top edge of the building (see Figure 35).

[1] Accuracies reported by architectural firm responsible for building design and general contracting in building reconstruction.

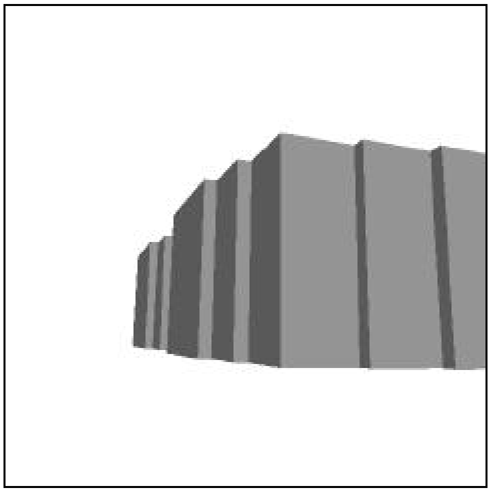


Figure 35. Rendering of Hardymon model. The model does not contain any detailed surface structure.

Three digital video sequences at 720x480 resolution were acquired at different locations around the building. Single shots were selected from these video sequences for testing our algorithm on isolated images. The location of the images is shown in Figure 36. All cameras are at roughly eye level above ground.

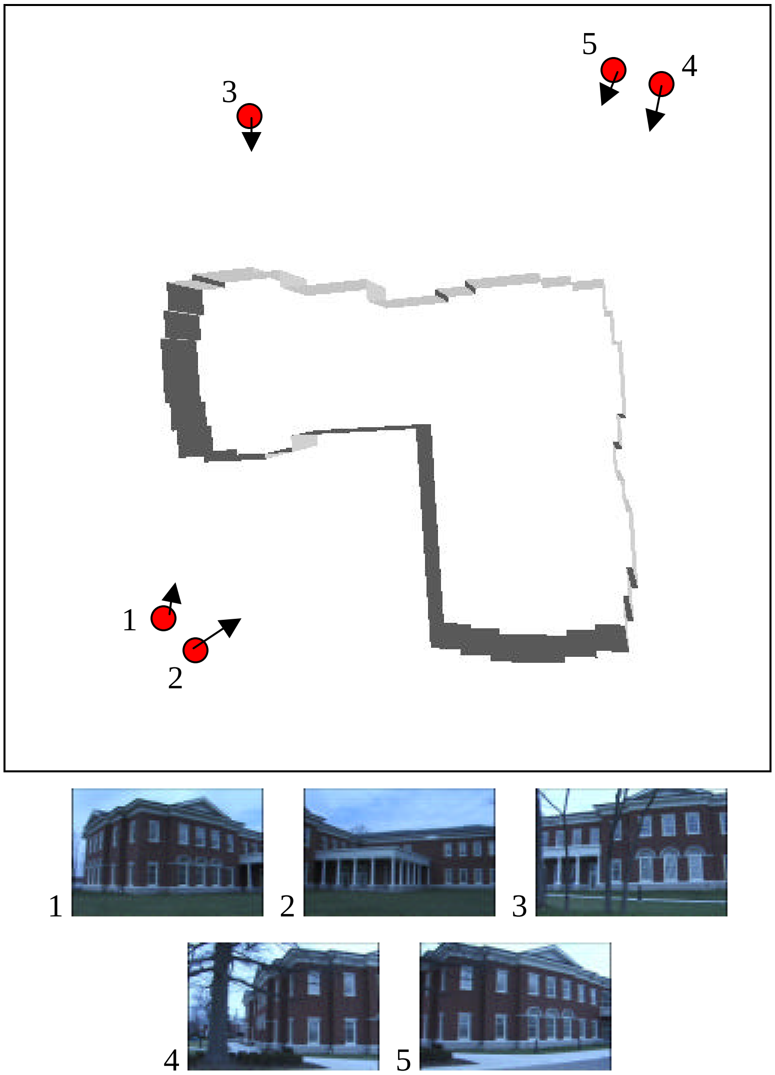


Figure 36. Top view of Hardymon scene with camera locations.

For each image, the initial camera parameters were supplied by a user who estimated both camera orientation and position using a simple interface that presents the model to the user in a graphical format.

The algorithm was then run for both datasets and the resulting camera parameters were compared to the initial parameters. In addition, the model was rendered with the final parameters. The results of these runs are shown in Section 6.4.

6.2 Line Finder Performance

The evaluation of the alignment algorithm begins with a simple analysis of the line finder used at the beginning of the algorithm. In order to evaluate the effectiveness of the line finder it is helpful to define the concept of *ideal lines*. Ideal lines are the lines in the image that correspond to model edges. The lines extracted by a perfect line finder would include all of the ideal lines. Measuring the percentage of ideal lines that are extracted by the line finder versus the number of ideal lines represented in the image serves to characterize the starting point of the remainder of the alignment algorithm. The line finder was run on all test images and the percentage of ideal lines extracted by the line finder was calculated. The results of this analysis are shown in Table 2.

Table 2. Line finder and facade extractor performance.

		House			Hardymon				
		1	2	3	1	2	3	4	5
Line Finder	Total Output Lines	970	1456	1071	278	206	412	296	359
	Visible Ideal Lines	15	10	16	25	11	9	44	42
	Ideal Lines Represented	13	10	12	18	8	8	21	19
	% Ideal Represented	87%	100%	75%	72%	73%	89%	48%	45%
	Concentration of Ideal Lines	1%	1%	1%	6%	4%	2%	7%	5%
Facade Extractor	Total Output Lines	22	57	26	28	32	33	27	36
	# Arising from Façades	21	49	20	26	31	29	23	30
	Concentation from Facades	95%	86%	77%	93%	97%	88%	85%	83%
	# that are Ideal	6	7	4	7	5	4	10	7
	Concentration of Ideal Lines	27%	12%	15%	25%	16%	12%	37%	19%
	Output Ideal / Input Ideal	46%	70%	33%	39%	63%	50%	48%	37%

Ideally, we would want the line finder to detect all ideal lines in an image. As shown in Table 2, this is not always the case. In most of the images, the line finder succeeded in detecting more than half of the ideal lines. For scenes containing many ideal lines, this should be sufficient. However, for scenes with few ideal lines, the failure to detect half of the lines could result in an inability to reach good alignment.

The concentration of ideal lines in the line finder output is fairly small. This indicates there is a significant amount of noise in the output of the line finder in the form of lines arising from the planar facade. This demonstrates the need for an algorithm that reduces the number of lines in the line set while preserving the ideal lines.

6.3 Façade Extraction Performance

The façade extractor performance was measured in several ways. The primary goal of the extractor is to detect building facades (see Chapter 4). Therefore, the output was analyzed for the percentage of lines that are related to building facades. An additional goal of the façade extractor is to reduce the number of building lines while preserving lines that correspond to ideal lines (see Section 4.2.3). The performance of the extractor in achieving this goal was measured by comparing the percentage of ideal lines versus total lines prior to and after the extraction algorithm. Every ideal line that is represented in the extractor input should also be represented in the extractor output if the extractor is performing optimally.

Table 2 also contains the results of the façade extractor performance for both the Hardymon and house scenes. The façade extractor dramatically reduced the number of lines in the test images while preserving façade structure. In all cases the concentration of lines arising from façade structure in the output of the extractor was greater than 70%, and in one case as high as 97%. This is especially meaningful given the small number of lines involved. In all but one case, the number of lines output by the façade extractor that were not related to a façade was six or less. In addition, the extractor favored ideal lines over other lines. Although image lines were reduced by greater than an order of magnitude by the façade extractor, the number of ideal lines was reduced by 67% in the worst case and less than 50% in three cases.

6.4 Image to Model Alignment Performance

An analysis of the iterative alignment step of the algorithm begins by characterizing the objective function on real data. Although results of the objective functions behavior on simple synthetic data are shown in Section 5.2, it is necessary to characterize the function on real data due to the complexity and noise introduced by imagery of real scenes. The objective function was sampled in a region of camera space centered on the aligned location. In order to better interpret and visualize the data, only two of the six external camera parameters were perturbed at any given sampling point. The results of the fitting function, over different values of the adjusted camera parameters, were graphed using 3D surface maps. The resulting surface maps were used to visually evaluate the quality of the objective function for alignment minimization. Figure 37 and Figure 38 show a resulting graph for the house and Hardymon datasets respectively. In Figure 37, the minimum value of the objective function lies at the aligned camera location, which is located at the center of the graph. However, the region around the minimum is monotonic within only a foot or so in the Y direction. There is also a false local minimum to the positive X direction of the aligned location. Hence, for this image and model alignment, the probability of finding a correctly aligned value decreases rapidly as error is introduced into the initial parameters. Note that although this graph provides only a 3D cross section of a six-variable function, and thus cannot be used to fully define the behavior of the function, it does provide insight into the function's behavior.

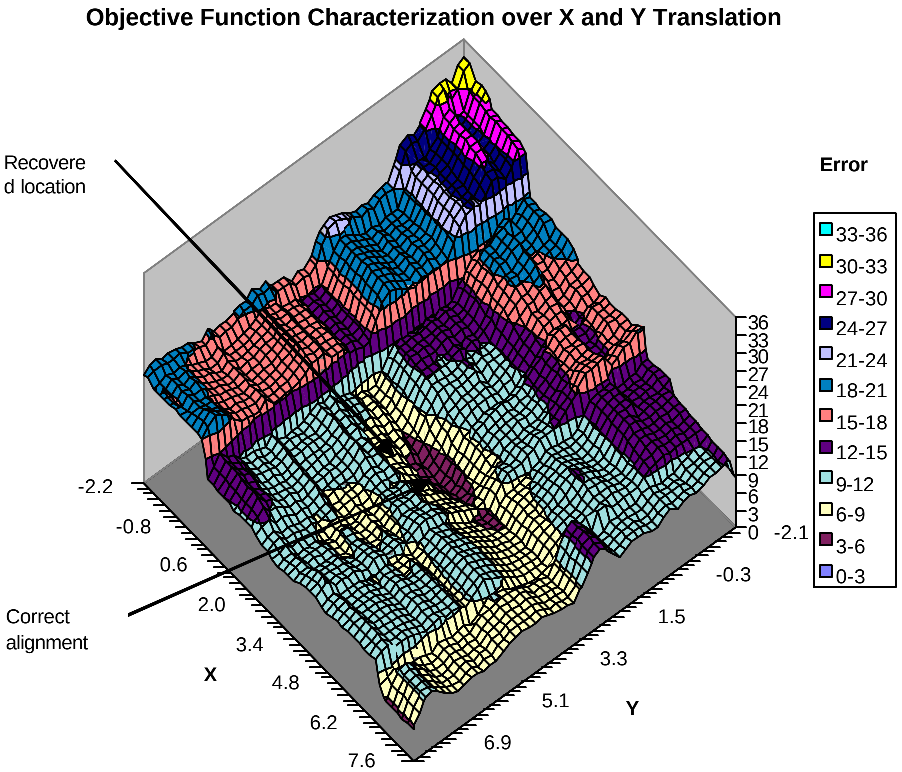


Figure 37. Objective function characterization over X and Y translation perturbation at the aligned camera location in house image 2. Cliffs in the function probably arise from locations in camera parameter space where a model line comes into or out of view. The correct alignment and the alignment recovered using filtered data are shown.

In Figure 38, the minimum value of the objective function is not located at the aligned location. This is probably caused by the fact that only 7 of 26 lines in the filtered line set are ideal lines (see Table 2). The remaining 19 lines are then noise that can pull the objective function minimum out of a true alignment.

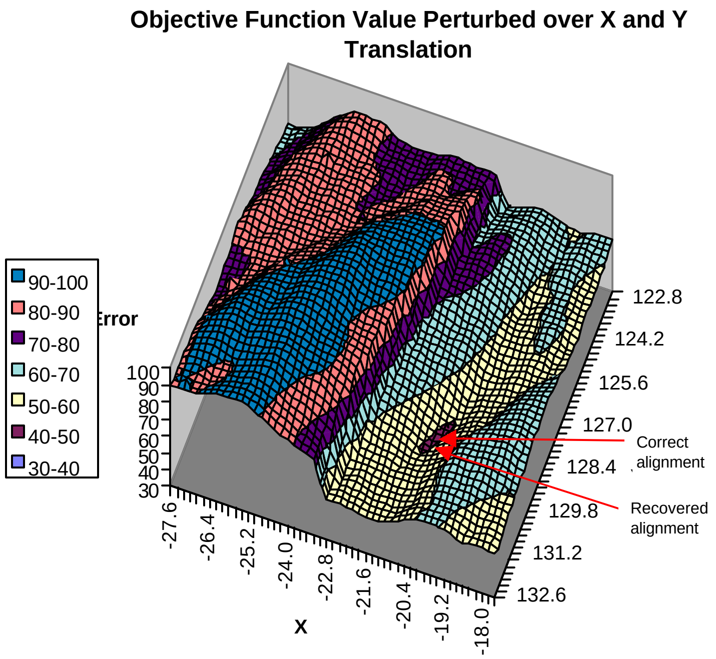


Figure 38. Value of objective function near aligned camera location for Hardymon image 1 perturbed over the X and Y transation parameters. The correct alignment and the alignment recovered using filtered data are shown.

Next, we consider the performance of the iterative alignment algorithm when using ideal line sets as input lines. Running the alignment routine on ideal lines is useful because it eliminates the effects of noisy data on the alignment. In other words, it measures the performance of the alignment minimization routine under the best possible conditions, thus setting an upper bound on the quality of the overall automatic alignment algorithm's performance. The ideal line set for each image was be extracted by hand by selecting lines in the images that correspond to model lines. The final objective function error for each alignment is shown in Table 3. Initial and final camera parameters are shown in Table 4. Renderings of the model using the initial camera location and the aligned imagery are shown in Figure 39, Figure 40, and Figure 41 to provide material for visual analysis.

Table 3. Final value of objective function when using exact lines and facade extractor output for alignment.

	FINAL VALUE OF OBJECTIVE FUNCTION	
Image	**Ideal Lines**	**Façade Extractor Output**
House 1	3.652	9.665
House 2	2.603	5.034
House 3	11.243	17.733
Hardymon 1	49.085	43.782
Hardymon 2	10.497	8.082
Hardymon 3	18.934	13.624
Hardymon 4	50.519	83.811
Hardymon 5	54.139	46.684

Table 4. Camera parameters for datasets. The initial (start), correct, and aligned camera parameters resulting from filtered and full line data are shown.

		Location			Orientation		
		X	Y	Z	X	Y	Z
House 1	start	-9	1	-52	2	-164	3
	filtered	-9.37009	0.803631	-52.94819	2.165603	-163.723	2.931383
	correct	-9.457621	0.714817	-53.01716	2.375017	-163.6628	2.707198
	full	-9.212412	1.156724	-52.1007	2.140459	-163.7621	2.464464
House 2	start	2	3	-61	-1	-167	1
	filtered	1.914438	3.82924	-60.48767	-0.450069	-166.6611	0.827767
	correct	3.497897	2.970762	-61.64317	0.179054	-168.8954	1.105556
	full	2.238652	3.221134	-60.73638	-0.092868	-167.1586	1.034427
House 3	start	69	8	39	-6	42	1
	filtered	70.01309	8.363791	38.38669	-6.534494	41.92584	0.28679
	correct	68.84719	8.830737	39.68284	-6.045843	41.43477	1.339421
Hardymon 1	start	-22.639	127.8483	24.29557	-81.40722	1.25538	30.47781
	filtered	-21.46845	128.8948	24.07372	-81.6534	-0.002766	31.01325
	correct	-20.89737	128.0853	24.225	-81.8596	0.797541	30.83661
Hardymon 2	start	-10.8081	145.5096	26.98523	-80.87163	1.87501	59.76706
	filtered	-9.415159	144.3373	26.27651	-81.98269	2.380739	60.82447
	correct	-10.39146	146.4511	26.16548	-80.91814	1.748588	59.58506
Hardymon 3	start	25	-56	23	99	181	-12
	filtered	24.26784	-55.50758	22.11371	98.60674	180.1858	-9.628523
	correct	26.87034	-59.14938	23.53913	98.59577	182.2099	-9.820246
Hardymon 4	start	177	-32	23	-80	1	199
	filtered	177.4686	-33.52274	23.89679	-79.16331	0.994464	199.6985
	correct	174.1489	-30.03685	23.66429	-79.65289	1.206049	197.6719
	full	176.7898	-31.95739	23.35282	-80.39786	0.975779	199.877
Hardymon 5	start	166	-32	24	-80	0	214
	filtered	166.0146	-31.27525	24.46902	-79.88201	-0.115651	214.1463
	correct	165.783	-31.94559	24.04988	-79.98226	-0.144614	213.6885

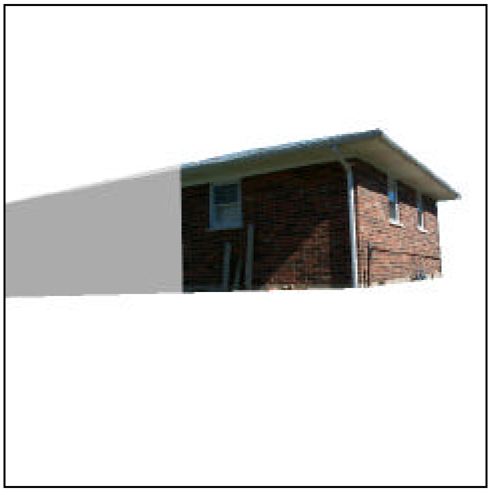


Figure 39. House model with House image 1 projected onto model after alignment using exact lines.

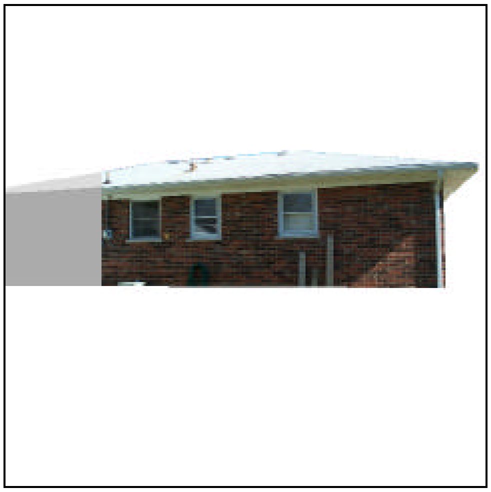


Figure 40. House model with House image 2 projected onto model after alignment using exact lines.

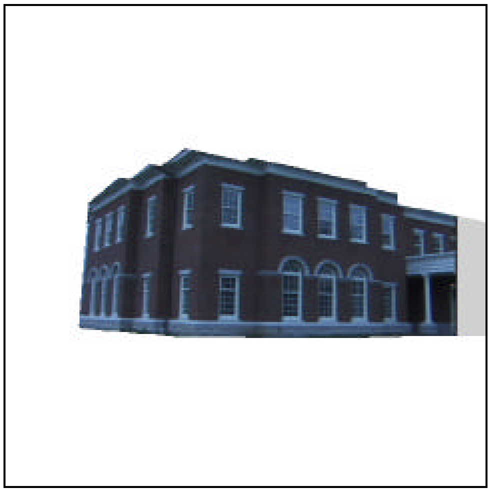


Figure 41. Hardymon model with Hardymon image 1 projected onto model after alignment using exact lines.

We now consider the performance of the alignment algorithm on filtered real data. Alignment results on several test images are shown below. For each image, we will explain why the image did or did not align well.

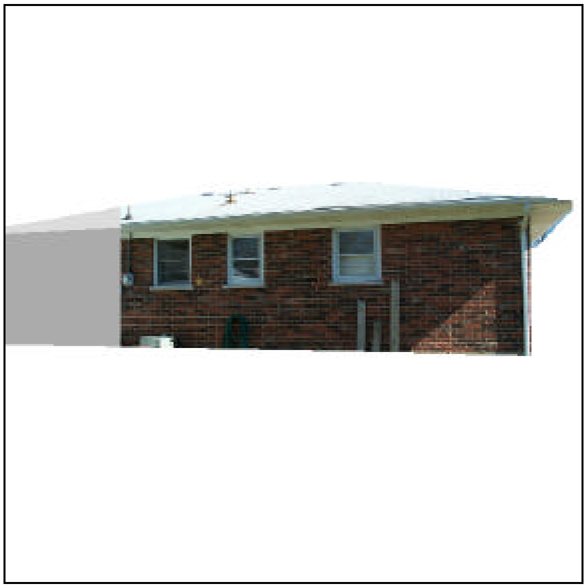


Figure 42. House image 2 projected from aligned position as determined from filtered lines.

Figure 42 shows the result of the alignment of image 2 of the house dataset. Note that a misalignment occurred along the bottom of the roof edge. The joint between the soffit and the fascia aligned with the top of the gutter, rather that with the bottom of the gutter. This is because of the overabundance of lines that were extracted in this region of the image (see Figure 43). With many local minima, it is easy for this type of misalignment to occur.

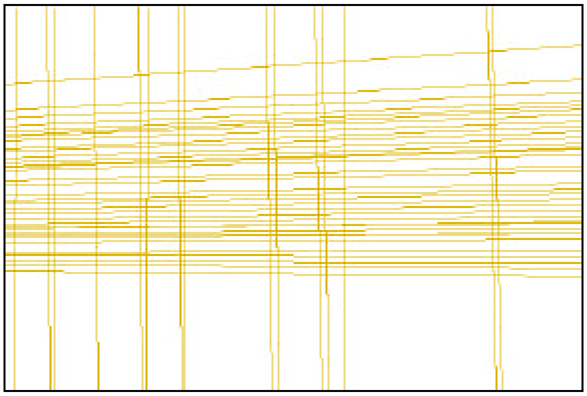


Figure 43. Lines extracted by the facade extractor on image 2 of the house dataset.

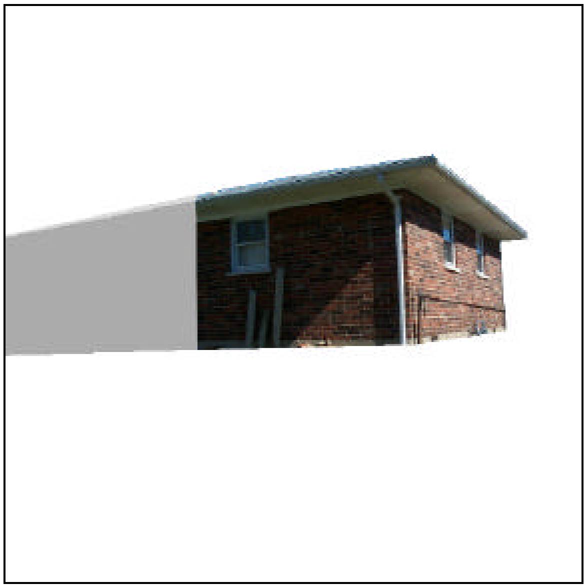
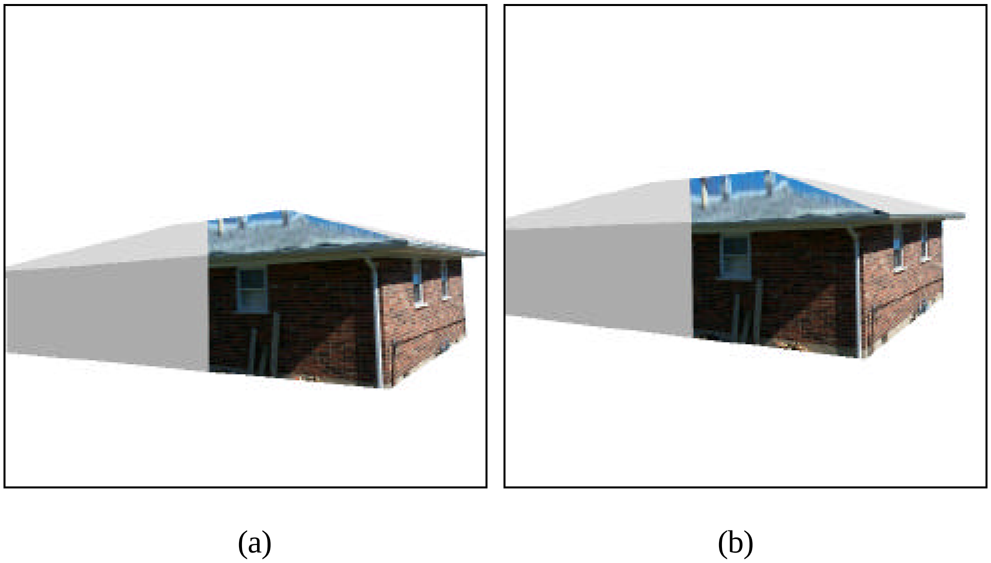


Figure 44. House image 1 projected from aligned position as determined from filtered lines.

Figure 44 shows the result of the alignment of image 1 of the house dataset. The house model stops with the brick; therefore, the bottom of the brick in the image should align with the bottom edge of the model. This did not occur because the line along the bottom of the brick on the right side of the house was not extracted clearly by the line finder. However, in spite of this misalignment, the model has a very realistic appearance. In particular, the roof aligned very well as evidenced by the natural appearance of the gutter projected onto the planar fascia of the model. The same alignment was run with the original, unfiltered lines and rendered in Figure 45. The model is rendered from a unique view to emphasize the differing results between running with filtered data and with the full line data.



(a) (b)

Figure 45. Alignment results using filtered (a) and unfiltered (b) data. The filtered alignment renders more realistically.

The roof line and brick line aligned better using the filtered data, in spite of the fact that the alignment error for the filtered data was 9.665 as opposed to 2.674 for the unfiltered data. The unfiltered data has a lower error due to the abundance of local minima arising from the line density throughout the image.

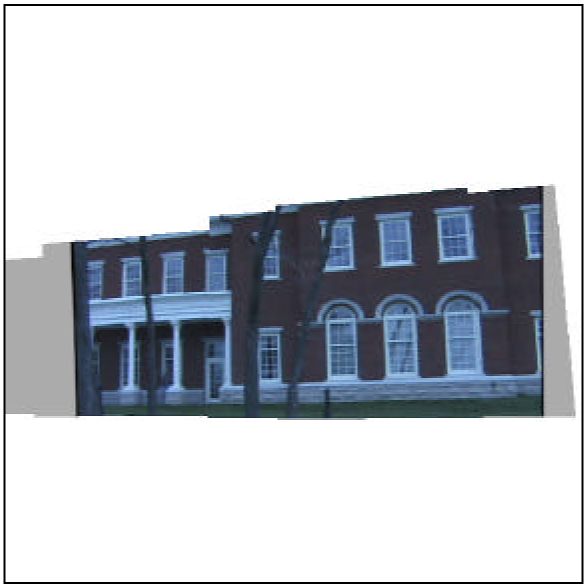


Figure 46. Hardymon image 3 projected from aligned position as determined from filtered lines.

Image 3 from the Hardymon dataset aligned very poorly (see Figure 46). As shown in Table 2, only four ideal lines were extracted by the facade extractor for this image. The remaining 29 lines that were extracted aligned falsely with model edges, resulting in a poor overall alignment. The number of ideal lines extracted by the facade extractor could be increased by utilizing higher resolution imagery with better lighting. In addition, incorporating more facade features in the model, such as windows, will result in more ideal lines interior to a surface that are likely to be extracted by the facade extractor (see Section 7.1).

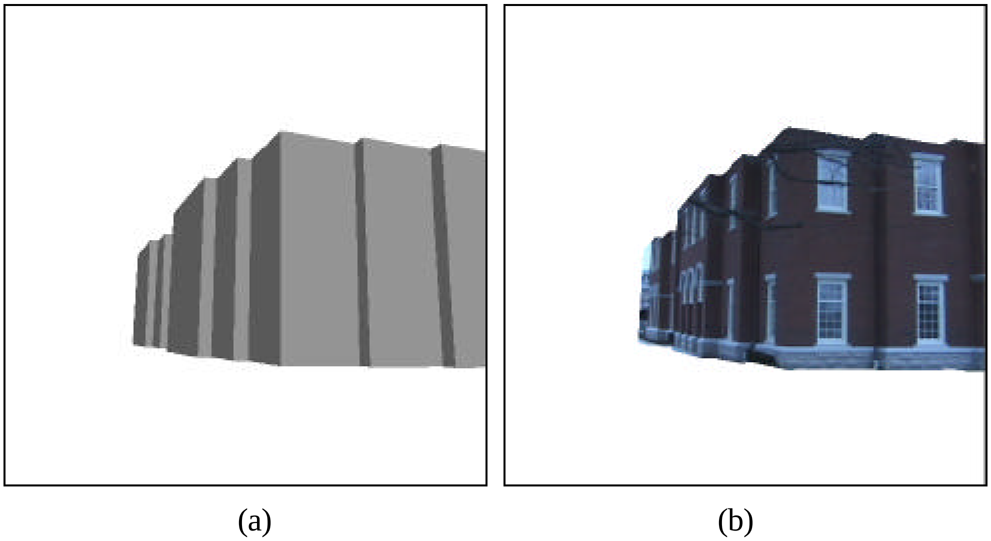


Figure 47. View of Hardymon model with (b) and without (a) projected image. Hardymon image 4 is projected from aligned position as determined from filtered lines.

The alignment of image 4 from the Hardymon dataset, shown in Figure 47 is visually compelling. Portions of the facade closest to the user are aligned with sufficient accuracy to appear to be a detailed 3D model of the structure. This image aligned well due to the high number of model lines extracted by the facade extractor (see Table 2).

6.5 Results

In order to better understand the convergence characteristics of the automatic alignment algorithm, the algorithm was tested to see how well it converges to the correct aligned position giving varying amounts of initial pose error. The algorithm was iteratively run after perturbing the pose information over a range of error amounts. At the conclusion of each iteration, the distance from the correct aligned parameters to the parameters calculated for the iteration was measured. We perturbed the pose information one parameter at a time over all parameters. A graph of one of the perturbation runs is shown in Figure 48.

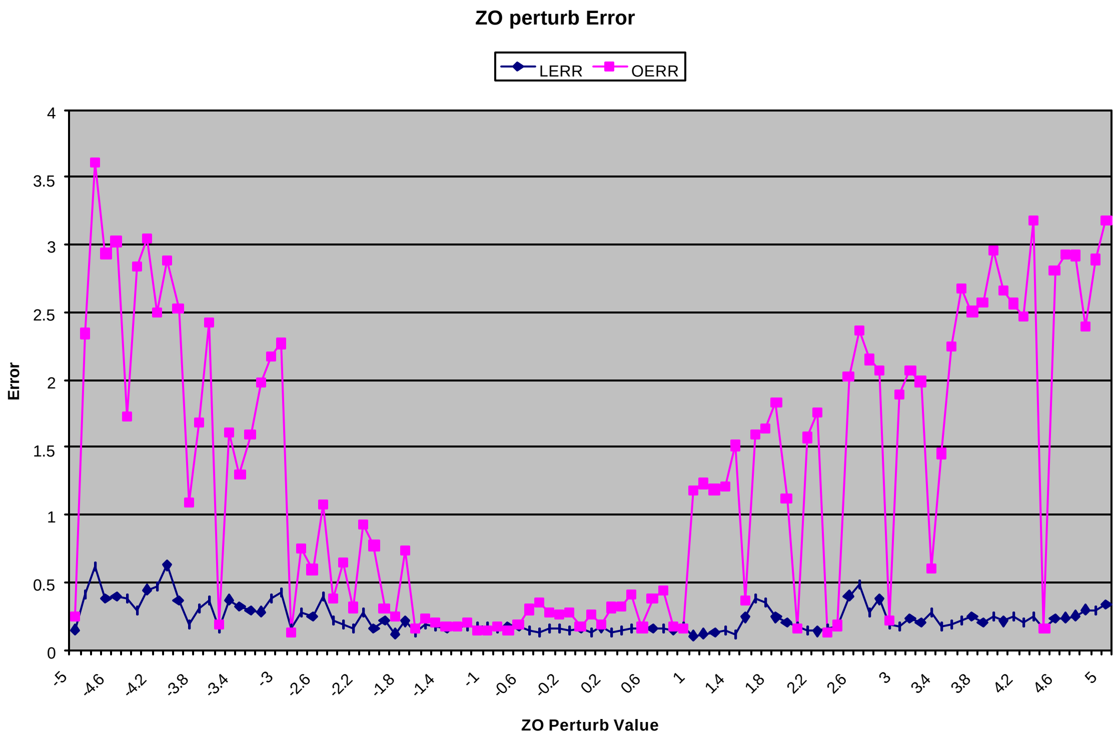


Figure 48. Error in final camera orientation (OERR) with perturbed Z rotation input. This chart shows that rotational orientation is recovered for Z rotation perturbations with 1 degree of the aligned location. The orientation error does not appear to have much impact on the final translation error (LERR).

This particular run was a perturbation of the Z orientation parameter. The resulting final translation error (LERR) is graphed in blue and orientation error (OERR) is graphed in purple. This graph shows that for this particular scene, the alignment algorithm was able to absorb between –1.7 feet and 1.0 feet of error in the Z orientation without adversely affecting the result of the alignment. Location alignment was not greatly affected by error in the orientation parameters. Similar results were obtained with perturbations of the other parameters (see Table 5). This data gives some indication of the convergence characteristics of the alignment algorithm. Due to differing levels of model complexity and orientations, the convergence ranges will likely vary across an image set. However, this data indicates that the algorithm is likely to converge if initial orientation parameters are within one to two degrees of correct alignment and if position parameters are within two to four feet of correct alignment. These requirements are within the accuracies provided by simple estimation and measurement techniques such as differential GPS.

Table 5. Alignment algorithm convergence analysis results. Each parameter was perturbed independently and the range of perturbation that yielded a good alignment is indicated. Parameters X, Y, and Z are in units of feet. Parameters XO, YO, and ZO are in units of degrees.

Perturbed Parameter	Recovery Range
X	(-2.0, 1.9)
Y	(-3.9, 3.1)
Z	(-2.9, 2.3)
XO	(-1.7, 0.7)
YO	(-2.9, 2.2)
ZO	(-1.7, 1.0)

Chapter 7
Conclusions

Several conclusions can be drawn from the data collected and analyzed in this thesis. In particular, there is sufficient evidence to prove the viability of the hybrid approach to automatic image to model alignment. The algorithm is shown to work effectively under certain conditions. The data also supports the conclusion that the weakest point in the alignment algorithm is the facade extraction step. Alignments are very successful when ideal lines are used as the alignment input line set. Nevertheless, the facade extractor does preserve a sufficient concentration of ideal lines to yield a good alignment in situations in which model boundary edges are likely to be detected by the line finder and not be confused by other near lines.

7.1 Future Work

Results in Chapter 6 show that good alignment can be expected as long as the facade extractor output contains a high concentration of exact lines. Therefore, techniques for enhancing the facade extractor output will have a significant impact on the overall performance of the algorithm. As described in Sections 4.2.2 and 4.2.3, such techniques can exploit the existence of approximate initial camera parameters to predict the location of vanishing points and facade boundary lines. In addition, models can be constructed to incorporate features that are likely to be extracted by the facade extractor.

Appendix A
Extensions of the Approach to Video

Although our main focus has been the development of a robust image-to-model alignment technique for single images, we introduce several natural extensions of this approach to video sequences. Video sequences have the desirable characteristics of dense coverage of camera parameter space as well as low variable in camera parameters between successive images. These characteristics can be exploited by the hybrid approach to both increase the quality of alignments and to decrease the amount of information that must be provided by the user as input to the algorithm.

A.1 Temporal Alignment Approach

The alignment minimization step of the automatic image to model alignment algorithm is designed to be robust within some relatively large neighborhood about the actual aligned camera parameters (see Section 6.4). If a relatively small upper bound can be assumed on the perturbation of the estimated camera parameters with respect to aligned parameters, then complexities associated with the alignment minimization algorithm can be reduced. For example, as described above, the input image data for the automatic alignment algorithm must be accompanied by approximate camera parameters. Although allowing approximations rather than exact parameters is a usability and data acquisition expense improvement, it is still a burden to the user. If it is known that a sequence of images were taken with sufficient temporal frequency to estimate total camera parameter variation between adjacent image pairs, then the need for approximate camera parameters for each image can be discarded. The reasoning for this conclusion is described as follows. As an agent moves through a scene, there is typically some bound on the speed at which the agent can change its viewing parameters. If the agent is a person, then this bound can be derived from normal walking speed and typical head movements.

Video sequences, which typically occur at frequencies between 20 to 30 frames per second, combined with typical human movement patterns, introduce a bounds on the image to image camera parameter variability. This bounds is within the convergence requirements of our automatic alignment approach. Hence, if a particular frame's aligned camera parameters are known, then these parameters can act as the estimated camera parameters for the next frame from the video sequence. Therefore, given only an approximate location for the initial image of a video sequence, the camera parameters for the entire sequence can be determined.

A.2 Vanishing Line Constraint

The existence of a sequence of images of the same structure, or a commonly oriented group of structures, allows for refinement of the aligned image parameters. As the vanishing points associated with each image are derived, they can be compared to the vanishing point locations from previous images to determine whether they fit the image sequence. The motivation for this approach arises from the observation that all lines in parallel planes with surface normal (A,B,C) will lie on a common "vanishing line" defined as $L = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \middle| Ax + By + Cf = 0 \right\}$ [42]. Thus, some subset of vanishing points of each image of a building or group of buildings will share a vanishing line. If the vanishing points of a new image do not lie on the vanishing lines as derived from other images, then it can be assumed that the new image is not aligned properly and the algorithm can take some action to correct the situation. Furthermore, after a number of images have been processed, the aligned camera parameters can be adjusted to minimize error of the vanishing points along the vanishing lines, leading to more accurate alignment results.

A.3 Preliminary Results

The video sequences described in Chapter 6 were used to test the ability of our algorithm to align successive images using the temporal alignment approach described in Section A.1. For each video sequence, an aligned set of camera parameters were provided for the first image in the sequence. The computed camera parameters for each image were then used for the subsequent image.

In each sequence, the algorithm was able to maintain alignment for several frames before losing alignment beyond recovery. The alignment was lost when a particularly ill-suited image was processed yielding a poor alignment. This alignment was then beyond the ability of the algorithm to recover on the subsequent image. This problem could be solved using the following methods. First, the bad alignment could simply be dropped from sequence as if it had never occurred. The subsequent image would then be aligned using the aligned camera parameters of the most recent well aligned image. The vanishing line constraint approach described in Section A.2 could be used to test the quality of the alignment in addition to the raw objective function value provided at alignment termination. Alternatively, the algorithm could keep track of the cameras motion over time and use this motion to predict the aligned location of the next image in a sequence. If the result of the alignment algorithm did not follow the type of motion expected by the sequence processing algorithm, then recovery measures could be taken such as skipping images until a good alignment is found or by attempting to restart the motion estimation process.

REFERENCES

[1] Collins, R.T. and Jaynes, C.O. and Cheng, Y.Q. and Wang, X.G. and Stolle, F.R. and Schultz, H.J. and Hanson, A.R. and Riseman, E.M. “The UMass Ascender System for 3D Site Model Construction”, Radius97, pp209-222.

[2] Foerstner, W., "Mid-Level Vision Processes for Automatic Building Extraction," presented at Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images, Monte Vertia, Switzerland, 1995.

[3] Henricsson, O., "Analysis of Image Structures Using Color Attributes and Similarity Relations," in *Institute of Geodesy and Photogrammetry*. Zurich: Swiss Federal Institute of Technology, 1996.

[4] Lin, C. and R. Nevatia, "Building Detection and Description from monoculat aerial images," presented at ARPA Image Understanding Workshop, 1996.

[5] Haala, N. and M. Hahn, "Data Fusion for the Detection and Reconstruction of Buildings," presented at Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images, Monte Vertia, Switzerland, 1995.

[6] Fischler, A., T. H. Kolbe, and F. Lang, "Integration of 2D and 3D Reasoning for Building Reconstruction Using a Generic Hierarchal Model," presented at Sematic Modeling for the Acquisition of Topographic Information from Images and Maps, Bonn, Germany, 1997.

[7] Jaynes, C, Thesis. University of Massachusettes, 1998.

[8] Huertas, A. and R. Nevatia, "Detecting Buildings in Aerial Images," *CVGIP*, vol. 41, pp. 131-152, 1988.

[9] Wang, X., H. Schultz, E. Riseman, and A. Hanson, "Surface Microstructure Extraction from Multiple Aerial Images," presented at CVPR, San Juan, Puerto Rico, 1997.

[10] Wang, X., W. J. Lim, R. Collins, and A. Hanson, "Automated Texture Extraction from Multiple Images to Support Site Model Refinement and Visualization," presented at Computer Graphics and Visualization, Plzen, Czech Republic, 1996.

[11] Faugeras, O.D. and Robert, L. and Laveau, S. and Csurka, G. and Zeller, C. and Gauclin, C. and Zoghlami, I. “3-D Reconstruction of Urban Scenes from Image Sequences”. CVIU(69), 1998, n3, March, pp292-309.

[12] Teller, S. "Automated Urban Model Acquisition: Project Rationale and Status". DARPA98, pp 455-462.

[13] Debevec, Paul Ernest. Modeling and Rendering Architecture from Photographs. PhD Thesis, UCB 1996.

[14] Pollefeys, Marc. Self-calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences. PhD Thesis. Katholieke Universiteit Leuven, 1999.

[15] Fitzgibbon, A.W. and Zisserman, A. "Automatic camera recovery for closed or open image sequences". ECCV98.

[16] Koenderink, J.J., and van Doorn, A.J., "Photometric Invariants Related to Solid Shape," *Optica Acta*(27), No. 7, 1980, pp. 981-996.

[17] Ramachandran, V.I., "Perceiving Shape From Shading," *SciAmer*(259), No. 2, 1988, pp. 76-83.

[18] Blake, A., Zisserman, A., and Knowles, G., "Surface Descriptions from Stereo and Shading," *IVC*(3), No. 4, 1985, pp. 183-191

[19] Liow, Y.T., and Pavlidis, T., "Use of Shadows for Extracting Buildings in Aerial Images," *CVGIP*(49), No. 2, February 1990, pp. 242-277

[20] Lee, K.M., Kuo, C.C.J.[C. C. Jay], "Direct Shape from Texture Using a Parametric Surface Model and an Adaptive Filtering Technique," *CVPR*98(402-407)

[21] Witkin, A.P., "Recovering Surface Shape and Orientation from Texture,"*AI*(17), 1981, pp. 17-45.

[22] Ikeuchi, K., "Shape from Regular Patterns," *AI*(22), No. 1, 1984, pp. 49-75.

[23] Moerdler, M.L., "Multiple Shape-from-Texture into Texture Analysis and Surface Segmentation," *ICCV*88(316-320).

[24] Woodham, R.J., "Analyzing Curved Surfaces Using Reflectance Map Techniques," *MIT-AI*79(161-182).

[25] Higuchi, K.[Kazunori], Hebert, M.[Martial], Ikeuchi, K.[Katsushi], "Combining Shape and Color Information for 3D Object Recognition," CMU-CS-TR-93-215, December 1993.

[26] Babu, M.D.R., Lee, C.H., and Rosenfeld, A., "Determining Plane Orientation from Specular Reflectance," *PR*(18), 1985, pp. 53-62.

[27] Eom, K.B., "Shape Recognition Using Spectral Features," *PRL*(19), No. 2, February 1998, pp. 189-195.

[28] Ribeiro, E., Hancock, E.R., "Detecting Multiple Texture Planes using Local Spectral Distortion," *BMVA*00.

[29] Hoogs, A. and Bremner, W. and Hackett, D. “The RADIUS Phase II Program”. DARPA97, pp 381-400.

[30] Hoogs, A. and Hackett, D. and Barrett, T. “Image Understanding at Lockheed Martin Valley Forge”. DARPA97, pp 455-464.

[31] Beardsley, P.A. and Torr, P.H.S. and Zisserman, A. "3D Model Acquisition from Extended Image Sequences". ECCV96, pp II;683-695.

[32] Antone, M.E. and Teller, S. "Automatic Recovery of Relative Camera Rotations for Urban Scenes". CVPR00, pp II;292-289.

[33] Coorg, S. and Teller, S.J. "Extracting Textured Vertical Facades from Controlled Close-Range Imagery". CVPR99, pp I:625-632.

[34] Becker, Shawn and Bove, V Michael Jr. “Semiautomatic 3-D model extraction from uncalibrated 2-D camera views”. SPIE, Vol 2401, pp. 447-461, Feb 8-10, 1995.

[35] Pollefeys, M. and Koch, R. and van Gool, L.J. “Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters”. ICCV98, pp90-95.

[36] Taylor, C.J. and Debevec, P.E. and Malik, J. “Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach”. UCB, 1996.

[37] Barnard, S, "Interpreting Perspective Images," *Artificial Intelligence* 21:435-462, 1983.

[38] Brillault-O'Mahony, B., "New Method for Vanishing Point Detection," CVGIP(54), No. 2, September 1991, pp. 289-300.

[39] Mclean, G. F., Kotturi, D., "Vanishing Point Detection by Line Clustering," PAMI(17), No. 11, November 1995, pp. 1090-1095.

[40] Gamba, P., Mecocci, A. Salvatore, U., "Vanishing Point Detection by a Voting Scheme," ICPI(96).

[41] Shufelt, J. A., "Performance Evaluation and Analysis of Vanishing Point Detection Techniques," *PAMI*, 21(3), March 1999.

[42] Haralick, Shapiro, *Computer and Robot Vision*, Addison-Wesley, 1993, TA1632.H37.

[43] Bremner, W. and Hoogs, A. and Mundy, J.L. "Integration of Image Understanding Exploitation Algorithms in the RADIUS Testbed". ARPA96, pp255-268.

[44] Nagao, M. and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*. New York: Plenum Press, 1980.

[45] Tavakoli, M. and A. Rosenfeld, "Building and road extraction from aerial photographs," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 12, pp 84, 1982.

[46] Herman, M. and T. Kanade, "Incremental Reconstruction of 3D Scenes from Multiple, Complex Images," *AI*, vol. 30, pp 289-341, 1986.

[47] Forstner, Wolfgang. “3D-City Models: Automatic and Semiautomatic Acquisition Methods”. Photogrammetric Week '99.

[48] Liebowitz, D. and Zisserman, A. “Resolving Ambiguities in Auto-Calibration”. Royal, A-356, 1998, pp 1193-1211.

[49] Lowe, D. G., *Perceptual Organization and Object Recognition*. Boston, MA.: Kluwer Academic Press, 1985.

[50] Lowe, D. G., “Three-dimensional Object Recognition from Single Two-dimensional Images,” *AI,* vol. 31, 1987.

[51] Boldt, M. and Weiss, R. and Riseman, E.M., “Geometric Grouping Applied to Straight Lines”, *CVPR86,* 1986.

[52] Boldt, M. and Weiss, R. and Riseman, E.M., “Token-Based Extraction of Straight Lines”, *SMC,* no. 6, 1989.

[53] McLauchlin, Phil and Rahimi, Ali, “Horatio Image Processing Package”, available at http://www.ee.surrey.ac.uk/Personal/P.McLauchlan/horation/html/index.html.

[54] Lowe, D.G. and Binford, T.O., "The Perceptual Organization of Visual Images: Image Segmentation as a Basis for Recognition", *DARPA83*.

[55] Collins, R.T., and Beveridge, J.R., "Matching Perspective Views of Coplanar Structures Using Projective Unwarping and Similarity Matching," *CVPR*93.

[56] Segal, Mark and Akeley, Kurt, "The OpenGL Graphics System: A Specification (Version 1.2.1)", Silicon Graphics, Inc., 1999.

[57] Brunn, A. and Gulch, E. and Lang, F. and Forstner, W. “A Hybrid Concept for 3D Building Acquisition”. PandRS, v53, n2, 1998, April, pp119-129.

Vita

Mike Partington was born on September 28, 1973, in Anchorage Alaska. He graduated summa cum laude from the University of Kentucky in 1998 with a bachelor's degree in Computer Science. He worked for two years as a development engineer for Lexmark, International, and is currently working for Cypress Semiconductor as a CAD engineer.

Mike Partington