Xactware Solutions, Inc.

EXHIBIT 1004

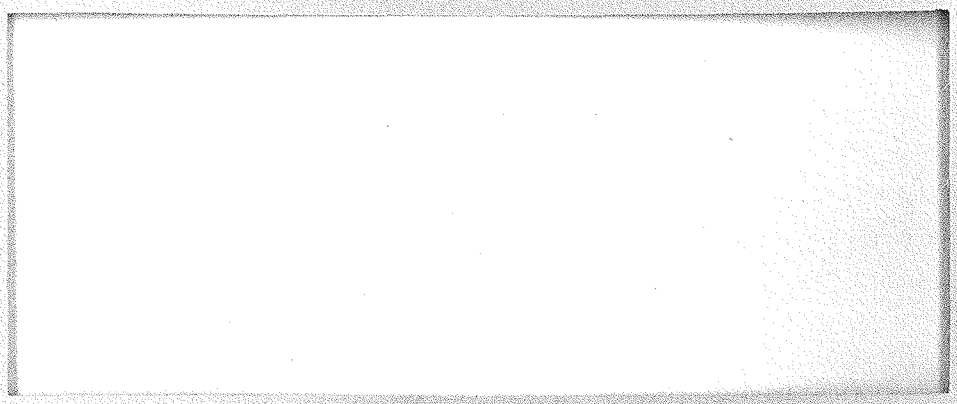
Xactware Exhibit 1004
Inter Partes Review of
US Patent No. 8,078,436

Design and Evaluation of a Semi-Automated Site Modeling System

Yuan Hsieh

November 1995
CMU-CS-95-195

Carnegie Mellon



Design and Evaluation of a Semi-Automated Site Modeling System

Yuan Hsieh

November 1995
CMU-CS-95-195

Digital Mapping Laboratory
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891

Abstract

This paper describes the design and evaluation of SITECITY, a semi-automated building extraction system integrating photogrammetry, geometric constraints, image understanding and user interfaces. Existing automated building extraction systems often produce mixed results and it is clear that human intervention is required to correct mistakes from fully automated systems. SITECITY gives human operators the ability to construct and manipulate three dimensional building objects using multiple images. Image understanding algorithms are integrated into SITECITY to assist users. These automated processes are selected and integrated based on methods and criteria derived from the GOMS (Goals, Operators, Methods and Selection Rules) human-computer interaction principle. The performance of these automated processes are rigorously evaluated. Twelve human subjects were used to evaluate the usability of the semi-automated system in comparison with the fully manual version. The methodology for the usability evaluation is described and the result of this evaluation shows that automated processes in SITECITY enhance the overall usability of the system and reduce the complexity of manual mensuration of three dimensional building objects.

This work was sponsored by the Advanced Research Projects Agency under Contracts DACA76-91-C-0014 and DACA76-95-C-0009 monitored by the U.S. Army Topographic Engineering Center, Fort Belvoir, VA. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency, the U.S. Army Topographic Engineering Center, or the United States Government.

Keywords: semi-automated computer vision systems, digital cartography, building detection, building delineation, human-computer interfaces, GOMS model, distance transforms, template matching, photogrammetry, performance evaluation, usability analysis

Contents

1 Introduction

Efficient and accurate delineation of man made features from digital aerial imagery has been a shared goal of the photogrammetry and computer vision communities for a number of years. The proliferation of digital photogrammetric workstations shows exciting possibilities for automating tedious and time consuming menial tasks. Among these tasks are automatic aerial triangulation, image mosaicing, and the generation of orthophotos and digital elevation maps [24; 20; 6; 31].

There currently exist three approaches to perform digital feature extraction tasks: fully manual, fully automated and semi-automated, all of which require images and camera parameters as input. A fully manual system requires users to measure all points on all building objects in all available images. On the other hand, a fully automated system requires no manual intervention and producing an accurate and complete site model in which users are not required to correct any mistakes. A semi-automated system requires users and automated processes in the system to work sequentially. For example, a semi-automated site modeling system can require users to supply the automated processes with inputs and cues. Based on these inputs, automated processes produce a site model, and finally, users correct mistakes in that site model.

Research in automated urban feature extraction, such as buildings and roads, has produced mixed results [17; 26; 33; 42; 12; 32; 25; 28; 36; 51; 54; 18; 2]. No system has been shown to work both accurately and completely with a variety of images and objects, and few [51; 36] produce 3D descriptions of extracted buildings in object space. In order to utilize these automated processes, an integrated system is needed that allows a user to correct and enhance the results of the automated systems.

While most research has focused on fully automated feature extraction systems, there are some systems which attempt to integrate manual and automatic processes [22; 48; 43; 19; 55; 24], where users and automated systems work in harmony. In [24], Heipke presented a survey of interactive softcopy photogrammetric workstations, which discusses the interactive and automated aspects of these workstations. Tilley [55] describes some current and potential uses of softcopy workstations and some automated extraction tools for natural feature data. In [19], a conceptual framework and applications of image understanding tools are described.

There are two recent examples of semi-automated building extraction systems. In [43], Mueller and Olson use a model based approach to detect and delineate rectangular and peak roof building models. In their system, users are required to specify a range of plausible parameters for the building models and approximate locations of the buildings in the image. The other system is the Cartographic Modeling Environment (CME) [22; 48]. It has evolved through the years to become a test bed for image understanding algorithms [44] and a site modeling system. It uses model-based optimization techniques such as snakes [30] to perform feature extraction in an interactive environment.

The concept of integrating automated processes in an interactive environment is a familiar one. The idea is to use the automated processes to simplify the tedium of manual tasks, and to give end users the ability to provide cues to reduce the problem domain for the automatic processes so that they will have a high success rate while reducing the amount of user interaction. However, there has been no formal discussion of methods to select, integrate and evaluate automated processes in an interactive system, especially when the automated processes are known to be imperfect. We believe that it is unlikely that perfect automated processes for man-made feature extraction will be developed in the near future, because digital cartography involves a large variety of imagery and scenes. It is difficult to design a system that can handle all possible cases. However, we believe that these imperfect automated processes can be rigorously integrated into a semi-automated system to increase the efficiency of cartographic extraction tasks. We have developed a system, SITECITY, that attempts to address these issues. In addition to proposing a design methodology for semi-automated systems, we propose a set of methods to evaluate the performance of the semi-automated system and to compare the usability of the semi-automated and the fully manual systems. These evaluations can answer the question: *How well does the system perform?* It cannot evaluate the *productivity* of the system, which depends on the goals and requirements of individual users, and we hypothesize an approach to determine the productivity of an interactive site modeling system.

SITECITY is a semi-automated three-dimensional site modeling system developed in the Digital Mapping Laboratory at Carnegie Mellon University. SITECITY is a tool for generation of sites using multiple images. A site is a collection of three dimensional man-made structures in a scene. SITECITY uses rigorous

photogrammetric principles and multiple images to accurately determine 3D locations of objects, such as buildings or roads in the scene. Photogrammetric methods supply cues to reduce the complexity of automated feature extraction tasks [39]. Automated processes such as model matching are used to reduce the number and complexity of manual measurements. A set of graphical user interface tools are provided for users to modify and create arbitrary objects and to supply other cues, such as area of interest, to assist the automated processes.

A snapshot of SITECITY is shown in Figure 1. Three windows (Figure 1(a), 1(b), 1(c)) are used for the image measurements and are called *image windows*. The fourth window(Figure 1(d)) displays the 3D position of the measured object and is the *site window*. Each image window has an identical graphical user interface in which a variety of menus provide support for measurement and modification of image features. Image features are triangulated to form 3D objects which are displayed in the site window for verification and visualization.

The main focus of this paper is the application of the semi-automated processes towards building detection and delineation in multiple images. First, the philosophical motivation for selecting and evaluating the usability of automated processes is described, followed by a description of various integrated automated processes for the construction of 3D building objects. Next, we propose a set of metrics to evaluate semi-automated systems and show the results of these metrics for SITECITY. These evaluation methods and metrics evaluate the accuracy and usability of the semi-automated system in comparison with the fully manual system, and can be adopted by other semi-automated feature extraction systems.

2 Methods and Criteria for Integration of Automated Processes

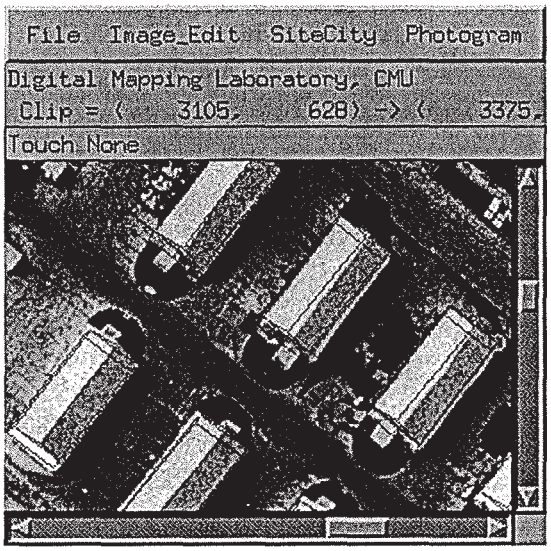
There are many interactive systems in fields other than computer vision and photogrammetry [47; 21; 9; 49] where *Intelligent Agents* or *Automated Processes* (APs) are used to assist users and to provide increased comfort and efficiency in their problem solving endeavors. While experimental results and performance analyses on some existing systems [52; 15] have been obtained, a discussion of formal methods to assess the performance and usability of these interactive systems has been lacking.

Common sense dictates that the inclusion and successful application of APs should improve the efficiency of the overall problem solving system. For some domains, APs can produce erroneous results; and these errors can be costly to users and decrease the overall efficiency of the system. When the cost of using APs exceeds some threshold, users will be not likely to utilize them. The problem is especially visible in the area of recognition-based user interfaces, such as speech recognition [49] and hand writing recognition [52], and in image understanding domains. In these fields, the state of the art is such that a perfect Intelligent Agent is not yet available. For speech and hand writing recognition, there is an emphasis on training the system and the user to bridge the gap. It is not clear if this approach is useful in an interactive image feature extraction system due to the diversity of the input data sets. It is also unclear if using an imperfect speech or handwriting recognition system is more efficient than using a keyboard and a well designed traditional graphical user interface.

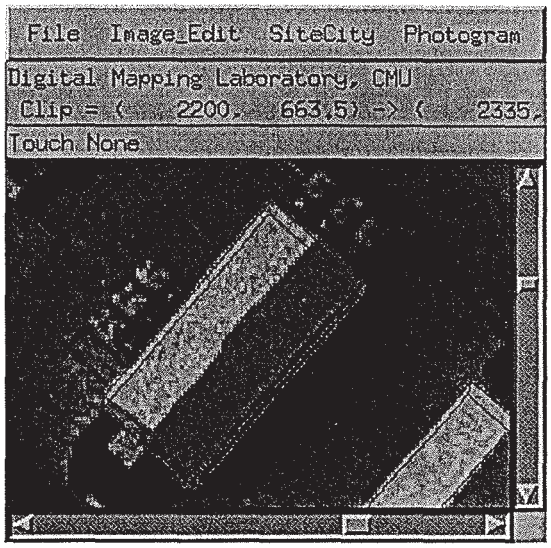
2.1 Cost of Problem Solving

The purpose of incorporating APs in an interactive system is to reduce the *user cost* of solving a problem. Conceptually, *user cost* can be defined by the *amount of work* required by the user to solve a problem. One way to quantify *user cost* is to use the Goals, Operators, Methods and Selection Rules (GOMS) model of human computer interaction [10; 29], which is an extension of the task decomposition strategies proposed in [45]. In the GOMS model, a problem (*goal*) can be decomposed into subgoals. To solve a problem, users select a *method*, which is a sequence of *operators* and *subgoals*, from a set of available methods based on a set of *selection rules*.

The GOMS model offers predictive power for estimating the efficiency and complexity of a human computer interface. The role of APs in the GOMS model is to solve subgoals or the entire goal for the user. Using the GOMS model allows us to analyze the efficiency of an AP by comparing the performance of the semi-automated system with the fully manual system. To perform this comparison, a task analysis approach [10] can be used. Ordinary, task analysis is used to predict human performance by decomposing a problem into

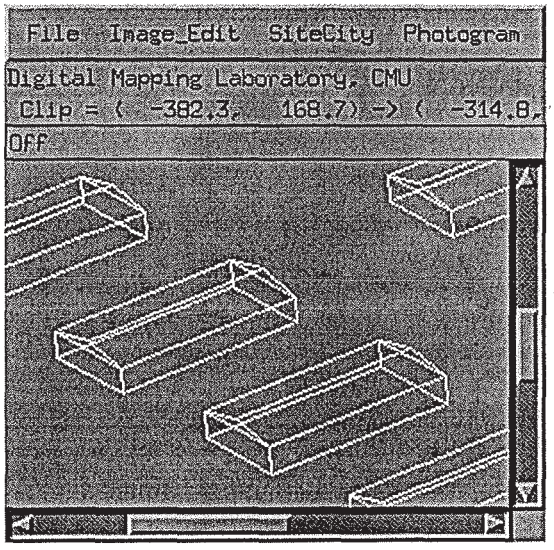


(a) Image measurement window: fhrad1



(b) Image measurement window: fhrad3

(c) Image measurement window: fhrad4



(d) Three dimensional display window

Figure 1: Snapshot of SiteCity

unit tasks. The elapsed time and the number of operations required to complete the tasks are collected. The human performance is evaluated based on the measured elapsed time and number of operations. The comparison of the human performance with and without using APs allows us to evaluate the utility of the automated processes in the semi-automated system.

A problem can be decomposed into a set of tasks. If every task is equally expensive to perform, in terms of the elapsed time, then the *user cost* of solving the problem can be considered to be the number of tasks the users need to perform. In this scheme, the role of APs is to complete a task or a set of tasks. For example, assume we are given a problem with N tasks. The *user cost* of manually solving the problem is N, because users are required to perform all N tasks. If an AP exists to perform one of these tasks, and the AP was able to complete the task correctly, the *user cost* becomes $N - 1$. However, if the AP does not perform the task correctly, the user faces the task of correcting the error. Errors of this nature can cause the *user cost* to be greater than N. A fully automated system which can solve all the tasks of a problem correctly will have a *user cost* of zero. This definition of *user cost* does not take into account the time to solve the problem, or the mental or psychological perceptions of solving the problem. Time is system and machine dependent; mental and psychological perceptions depend on the user interface and the users themselves.

Obviously, the measure of *user cost* is system dependent, because different systems can decompose a problem into different sets of tasks. For example, consider the problem of drawing a square. In one decomposition of the problem, the tasks are measuring the xy locations of 4 points, because a square must have 4 points. The property of equal length on every edge of a square is a constraint which these 4 points must satisfy and is not considered to be a task. A different decomposition might define the square parametrically by its xy position, side length, and orientation, and three tasks will be required to draw a square. It will be difficult to directly compare these two drawing programs because the cost of measuring a point's location might be different than the cost of specifying a parameter such as orientation. Therefore, to readily compare the performance of a semi-automated system with a fully manual system, the two systems must decompose the problems in similar fashion.

Using the square-drawing problem as an example, the role of APs and methods for comparing the fully manual and semi-automated system can be illustrated. The fully manual drawing program requires measurement of four points and has a *user cost* of 4, assuming point measurement is a unit task. A different semi-automated drawing program is developed to take advantage of the constrained geometric shape of the square and only requires the input of 2 points (2 opposing corner points). The semi-automated program still needs to calculate the position of the remaining 2 points based on the input points so 4 tasks were still accomplished. However, the semi-automated drawing program used an AP to perform 2 of the 4 required tasks. If the semi-automated program does not make any mistakes, it will incur a *user cost* of 2, because only two measurements were performed by the user. The semi-automated drawing program will be an improvement over the fully-manual program, and can be considered usable. However, *user cost* alone does not truly measure the usability of the system, because elapsed time is not considered. If a user can manually measure a point in two seconds while the AP takes an hour to perform the same task, then it would be difficult for a user to accept the semi-automated system.

Therefore, in addition to comparison of the *user cost*, another component of the usability analysis is the comparison of elapsed time, which is not explicitly addressed by the *user cost* analysis. In the *user cost* analysis, we assume that each unit task can be solved in the same amount of time and that APs are assumed to require no time. In reality, APs require finite amounts of time that might be different from the times required by users. The time requirement of APs needs to be considered to determine the overall usability of the semi-automated system.

In our square-drawing problem, if the time required to measure the xy location of a point is t, then a user can draw a square in $4t$ units of time using the fully manual program. Assuming the semi-automated program is perfect, the time required by a user using the semi-automated program to draw the square is $2t + T$, where T is the time required for APs to determine a solution. If T is equal to $2t$, then a user using the semi-automated program will spend the same amount of time as the fully manual program, but fewer manual operations were needed in the semi-automated program. If T is less than $2t$, then the semi-automated program is a definite improvement over the fully manual program. However, if T is greater than $2t$, the usability of the system will be in doubt.

2.2 Criteria for Selecting Automated Processes

Not every automated algorithm is suitable in an interactive system; some algorithms are more usable than others. Commonly, the *usability* of a system is defined to be the *ease and effectiveness* of the system. Ease and effectiveness include concepts such as flexibility, learnability, error handling, speed and accuracy [11; 49; 53].

A usable AP can enhance the usability of the entire system by decreasing the amount of work. A perfect AP that does not decrease the *ease and effectiveness* of the system will always be usable. At what point will an imperfect AP be usable? We argue that an imperfect AP is *potentially usable* if it reduces the overall *user cost* of the system, and *usable* if it reduces both the *user cost* and the elapsed time of a user. A *potentially usable* AP reduces the number of manual tasks for a user at the expense of time. If the time requirement is great, then it is unlikely that the AP will be used and cannot not be considered *usable*. However, the AP can be re-engineered to be more efficient, or advances in hardware can reduce the time requirement for the AP, and it is possible that the AP can become *usable* in the future. Based on these definitions of *usability* and *user cost* , four criteria are proposed to determine the feasibility of an automated algorithm in an interactive system:

- **Speed and Robustness**

  A fast algorithm will reduce the idle time of the user. A robust algorithm should not be affected by differences in scenes, images, object types, and users. Of the two criteria, robustness should be more important. Advances in computer hardware will make some slow algorithms practical. An unreliable algorithm will remain so regardless of hardware and should not be integrated into an interactive system.

- **Non-Intrusiveness**

  Automatic processes must not be a hindrance to the user. If an automated process fails in its tasks, the failure should not add any *user cost* to the task. Let the *user cost* of solving a problem without an AP be A, and solving the same problem with an AP be B. If for all scenes, the average value of A is greater than the average value of B, then the AP should be considered usable. Otherwise, the inclusion of the AP should be reconsidered. For example, consider a building detection algorithm, M, which returns 20 false building hypotheses for an image where there is only one building. For the same image, algorithm N returns nothing. Both processes failed to detect the building correctly. However, at the end of process M, a user must visually verify and delete all the false hypotheses, in addition to manually measuring the building; whereas users of process N only need to measure the building. Everything else being equal, process N should be considered superior to process M for integration in the semi-automated system.

  In addition, a user should not need to alter problem-solving strategies when using the automated processes. Once the tasks have been decomposed for a problem in a system, the users should be able to rely on the same strategy to solve the problem, with or without the use of automated processes.

- **The Janitor Model**

  Some APs are governed by a set of parameters. Optimizing the performance of these APs often becomes a problem of finding the optimal combination of these parameters. Often, these optimal values are problem and task dependent. Some of these parameters are intuitive; others are only meaningful to the developer of the algorithm. These parameters contribute to the overall complexity of the system and decrease the learnability.

  We believe that it is unreasonable to expect an average user to understand the theoretical intricacies of an AP in order to optimize the performance of the AP within the interactive environment. For a system with a set of complicated parameters, it is likely that the time it takes for a user to find a set of plausible parameters is better spent performing the task manually. The act of *parameter tweaking* also violates the non-intrusive criteria. Therefore, we argue that an interactive system should be designed according to the *janitor model* [40]; i.e., a janitor should be able to sit down and perform the task. Clearly, the janitor may need some minimal training, but the level of training is commensurate with the complexity of the tasks.

In a survey of CAD users [15], it was found that users tend to have a small command repertoire, and these users complained that commands and parameters were too abstract and inconsistent. Half of the subjects report that they felt lost in the system and felt it would be easier and faster to use pen and paper rather than CAD. These findings further support the idea that a simple system is a better system for human interaction.

- **Flexibility**

  Often, some APs will make many limiting assumptions in order to reduce the complexity of the problem. However, these limitations should not affect the operation of the interactive system. The inability of an algorithm to detect circles should not limit the ability of the user to draw and measure a circle. In addition, an AP that can be applied to different tasks and be reused can reduce the complexity and increase the maintainability of the system.

As with many system design problems, judging the feasibility of APs in an interactive system involves trade-offs. If an automated process is correct 99% of the time, the question to ask is: " Does the usefulness of the automated processes offset the cost of error correction and the complexity?" In section 4.6 we attempt to address this question for SITECITY using the proposed definitions of *usability* and *user cost* , bearing in mind that these measures can not take into account purely psychological factors; user preferences are difficult to predict.

3 Automated Processes in SiteCity

The criteria presented in Section 2.2 allow us to determine the feasibility of automated processes for integration into a semi-automated system. However, we still need to determine specific tasks in SITECITY to automate. Again, the task analysis approach can be utilized to determine the problem domains of the APs. In SITECITY, the design of the fully-manual interactive system dictates the task decomposition of a problem. Once the tasks are known, the APs can be selected and evaluated for integration. Currently, SITECITY is designed for detection and delineation of three types of buildings:

- **Flat Roof Building Model:** A flat roof building model has two horizontal polygons, the roof and the floor. These horizontal polygons are connected by a number of vertical rectangles, the walls. An example of a flat roof building model is shown in Figure 2(a). There are no limits on the number of vertices for the roof and floor polygon, but the roof and floor must have the same number of vertices.
- **Rectilinear Flat Roof Building Model:** A rectilinear flat roof building model is similar to a flat roof building model. However, it has one further constraint; the roof and the floor polygons must be rectilinear (right angles at every vertex). An example of the rectilinear flat roof building model is shown in Figure 2(b).
- **Peak Roof Building Model:** The peak roof model is shown in Figure 2(c). The bottom half of the peak roof model is a rectilinear flat roof building model with a rectangular roof and floor. The roof is a triangular prism.

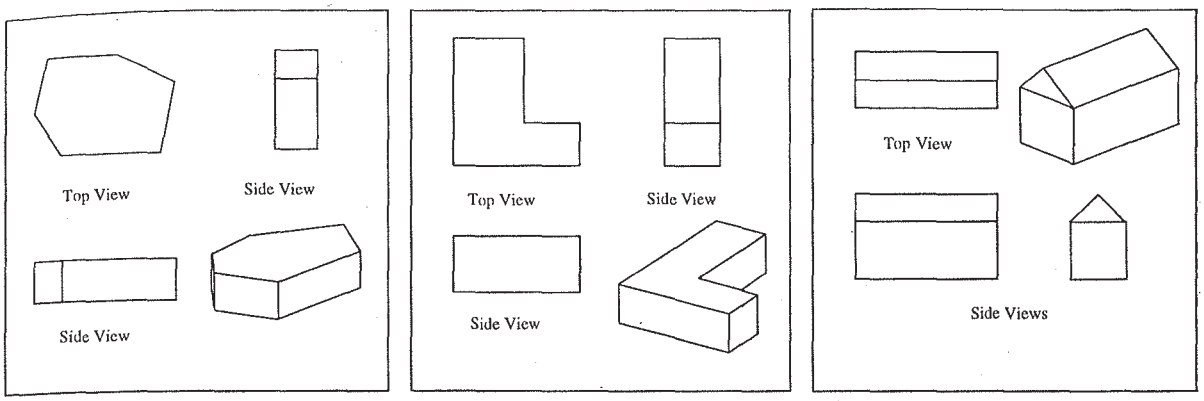
To create a 3D building object, the outline of the roof is measured and SITECITY estimates the dimensions of the building. An example of manual building creation is shown in Figures 3 and 4 and is as follows:

1. **Roof Measurement**

   The first task in SITECITY is to manually measure the roof of a building in an image. SITECITY creates a building model based on the delineation of the roof. The building model has a default height dimension of 30 meters, and is projected to all images and object space using a DEM to determine the elevation of building floor. (see Figure 3(a) and 3(b)).

2. **Floor Position Measurement**

   Since the building height was determined arbitrarily and most likely incorrect, the next task is to measure the floor position(Figure 4(a) and 4(b)).



(a) Flat roof building model

(b) Rectilinear flat roof building model

(c) Peak roof building model

Figure 2: Three building models

3. **Correct Building Location In Other Images**

   The final task is to correct the location of the building in other images (See Figures 4(c) and 4(d)). When an object is projected from one image to the next, the projected buildings usually do not agree with the image due to inaccuracies in the DEM and the camera parameters. This step is essential for achieving accurate 3D building models.

Given this procedure, two tasks were identified for automation.

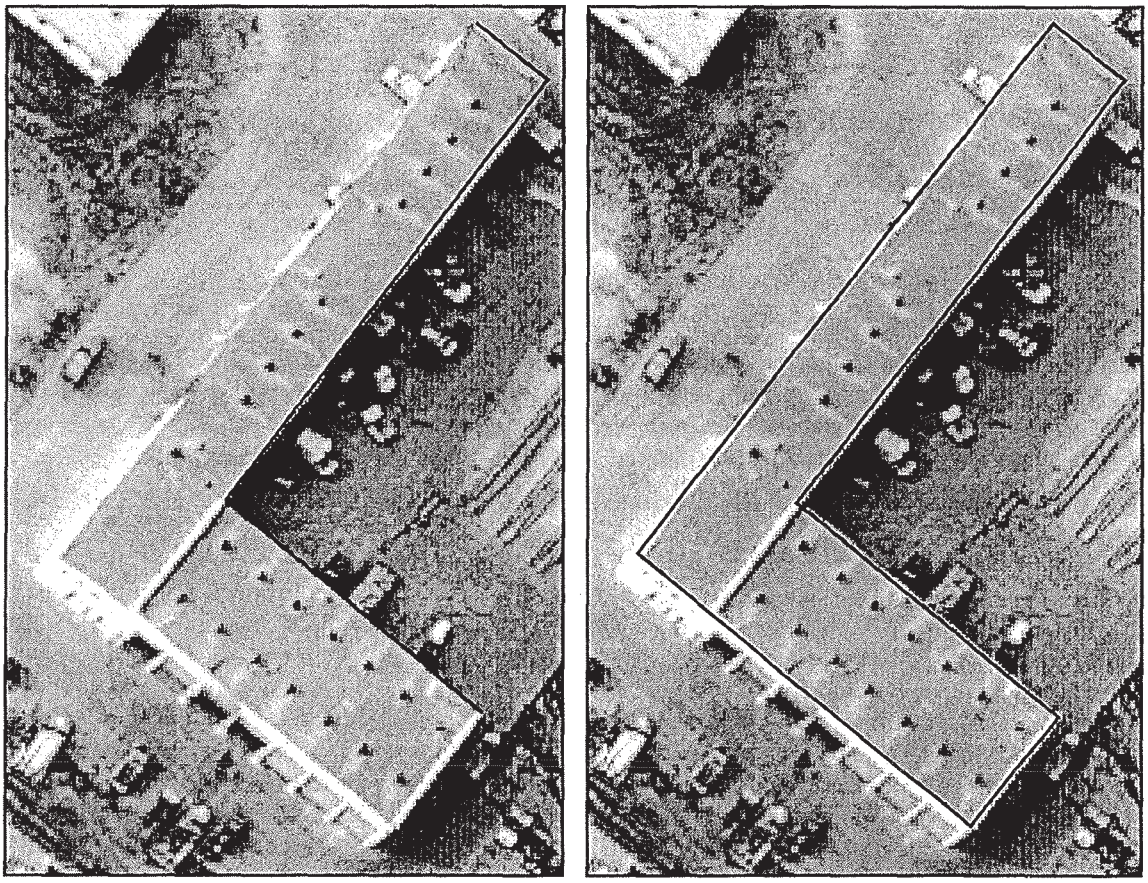
1. **Floor Position Measurement**

   Given a roof delineation in an image, the purpose of this task is to determine the height of the building for the roof using a single image. This task involves location of the floor, based on the roof delineation. A similar task was performed in [36]. This task should be non-intrusive. If the automated process for task 2 failed, it should not require the user to perform more operations than he otherwise would have had to. Without the automated processes, the floor position is set to a default value. With the automated processes, if the estimated floor position is erroneous, this position should be no worse than the default position, and the user would perform the floor position measurement tasks as if no automated processes were used. The only drawback is the addition of one system parameter, which bounds the search space for building height. Another attractive feature of automating this task is that the process can be used with both the flat roof and peak roof building model.

2. **Correct Building Location In Other Images**

   This task involves correcting the location of the building in other images (See Figure 4(c) and 4(d)), and is related to many stereo or multi-image matching processes in which image features were matched along the epipolar line [5; 38; 13; 51; 3; 14]. The relative success of the stereo processes make this task a likely candidate for automation. In addition, many current building detection approaches have also succeeded in using stereo to verify roof hypotheses [42; 18]. The APs for this task can also be designed to be non-intrusive; if the process failed, the user does not have to do any additional error correction, instead just performing this task manually. The additional complexity of this automation lies in two system parameters that bound the search space in terms of disparity range. The process can also be building object independent.

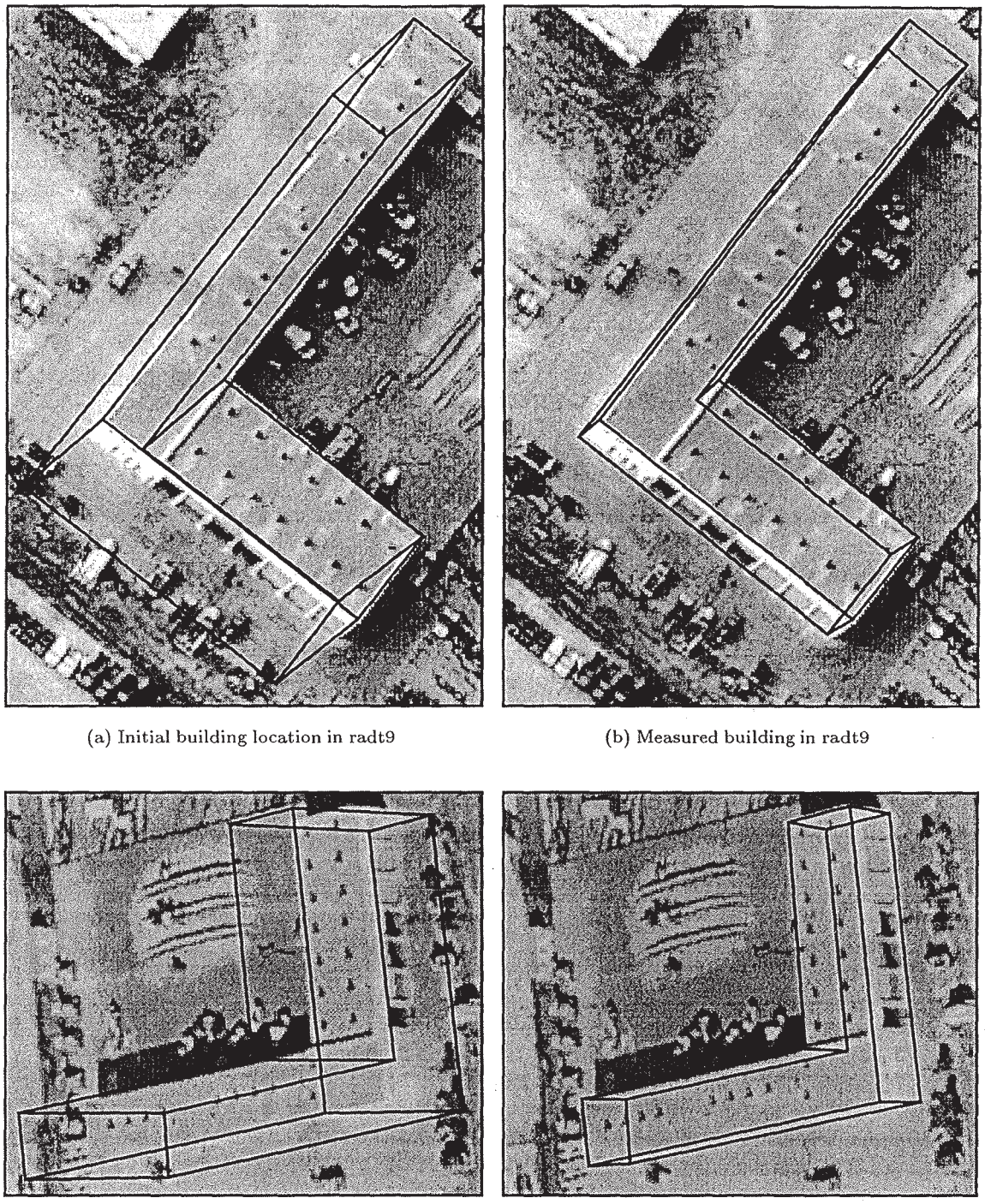
Figure 5 shows the *control* flow design of SITECITY. First, a user identifies a task, which can be performed using a set of manual processes. For some manual processes, automated processes are available to perform parts of the task. These automated processes shares three automated components.



(a) Measure the first four roof points in radt9 image

(b) Measured roof in radt9 image

Figure 3: Measuring the roof of a flat roof building in one image



(a) Initial building location in radt9

(b) Measured building in radt9

(c) Initial projection in radt9ob

(d) Corrected projection in radt9ob

Figure 4: Correcting building delineations

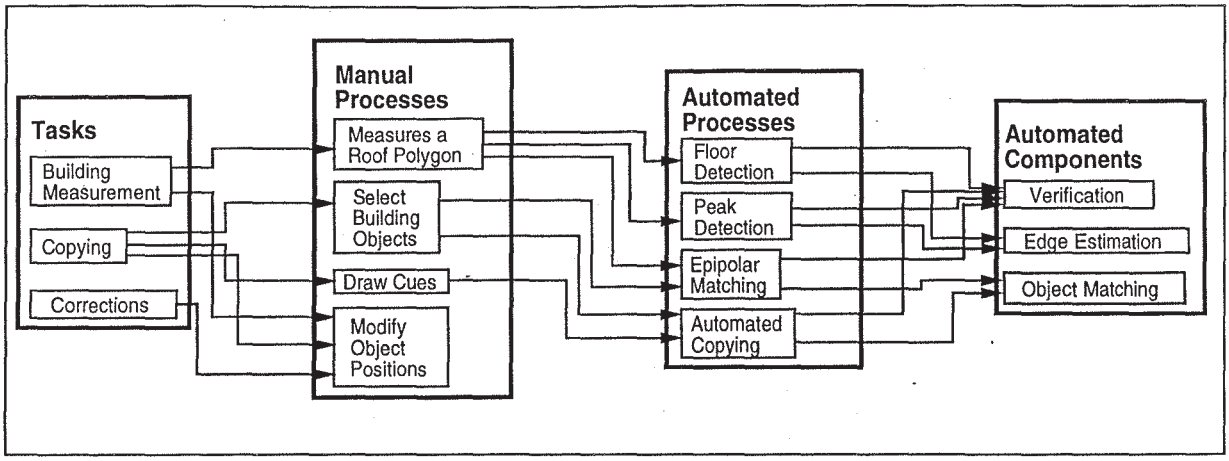


Figure 5: Control Process Flow of SiteCity

3.1 Components of Automated Processes

For the problem of building measurement, two tasks were identified as candidates for automation within the SITECITY system. These tasks described in the previous section can be broken down into three component processes; a verification component, an object matching component and an edge estimation component. The verification component takes an object in image space, and verifies the existence of the object using the image and the 3D geometry of the object. Given an object model in image space, the object matching component generates a set of hypotheses in image space for the object location in an image. The verification component can be used to select feasible hypotheses. The edge estimation component takes an model edge, with a known orientation in object space, and uses search region specified by the automated processes for matching. This component generates a set of possible edge hypotheses for the model edge.

3.1.1 Verification Component

The goal of the verification component is to verify the existence of a hypothesized building object in an image. In most existing fully automated building detection algorithms, the verification of building hypotheses are performed on the roof structures. Since most of these automated algorithms only hypothesize two dimensional roofs in image space, the verification of building hypotheses cannot take advantage of the 3D geometric and photometric constraints.

The verification component used in SITECITY uses full 3D geometry to predict the visible building and shadow structures. The task of verification is to match the hypotheses with image features. When a 3D object is projected to an image, some of the edges might not be visible due to the viewing geometry. Therefore, before any verification is attempted, the hidden lines of the object in the image are first removed from consideration. The result is a two dimensional line drawing of the expected visible edges of the object in the image, and the problem reduces to matching this 2D line drawing to the image. The shadow structure of the hypothesis is constructed using the knowledge about the sun's elevation and azimuth angles during image acquisition, and the viewing angle of the camera. The shadow is projected to the ground, which is assumed to be locally flat and horizontal with the same elevation as the floor of the building.

The verification component is based on the following functions:

- **Visible Edges**

  The visible edge function is similar to many edge template matching algorithms using distance transforms [27; 8; 46]. Let $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$ be the finite set of edge points in a building hypothesis and the image, respectively. The chamfer distance [7] of point a_i in A with respect

to B is defined as

$$C(a_i, B) = \min_{b \in B} ||a_i - b||.$$

The set of chamfer distances for all points in A can be computed,

$$CA = \{C(a_1, B), C(a_2, B), \ldots, C(a_n, B)\}.$$

Let $V(CA, v)$ be a function that returns the number of points in CA whose chamfer distance is less than v. v is a threshold that determines the maximum distance for a point in A to match a point in B. Then, V determines the number of points in A that have a match in B. Let m be the number of points in the model, then

$$P(CA, v, m) = \frac{V(CA, v)}{m}$$

is the percentage of points in CA that satisfies the distance threshold, v. In other words, P is the percentage of model points that match image points.

is defined as the chamfer score

The visible edge scoring component is defined to be

$$VE(A, B, v) = min(P(CA, v, m), P(CB, v, n)),$$

where

$$CB = \{C(b_1, A), \ldots, C(b_m, A)\}.$$

In SITECITY, the value v is currently set to 2 pixels, to account for uncertainty of the position of edge points and the discretization error of line segments.

- **Edge Gradient**

  The visible edge (VE) component is susceptible to the particular edge detection algorithm used to construct the set of edge points in the image because different edge operators have different responses on the same image. Furthermore, it does not account for the contrast across edge boundaries, which makes it sensitive to coincidental alignments of edge points. An edge gradient function is used to account for the contrast across edge points.

  Let A be the set of edge points in the model, $grad(I, x, y, \alpha, \beta)$ be the gradient of the image I, at position $[x, y]$, perpendicular to direction vector $[\alpha, \beta]$. Edge visibility is defined as:

$$EG(A, I) = \frac{\sum_{a \in A}(grad(I, x(a), y(a), \alpha(a), \beta(a)))}{|A|}.$$

  Intuitively, $EG(A, I)$ is the average contrast of the expected edge points in a model.

- **Visible Shadow Edges**

  Given a 3D building object, we can compute its expected shadow boundary in the image given the camera parameters, sun azimuth and elevation angle and the shadowed surface (the ground). Currently, SITECITY assumes that the shadowed surface is a horizontal plane lying at the same elevation as the floor of the building object.

  To compute the shadow region, each individual facet of the building is projected to the ground. The projected regions are merged into one polygon and the building regions are subtracted from the shadow region. Once the shadow boundaries are known, the ground/shadow boundary can be determined , and the visible shadow edges can be evaluated using $VE()$.

  Let S be a set of points on the shadow boundary separating the ground and the shadow, $S = \{s_1, s_2, \ldots, s_p\}$, and B be a set of edge points in the image, $B = \{b_1, b_2, \ldots, b_n\}$. The visible shadow edges component is defined as

$$VSE(S, B, v) = VE(S, B, v).$$

- **Shadow Color**

  In most cases, shadow regions should be darker than the ground on which they are cast. This observation can be used as an additional constraint. Let $G = \{g_1, g_2, \ldots, g_l\}$ be a set of points in the ground region that is less than T pixels away from the shadow-ground boundary (See Figure 6). G can be computed using the chamfer distance image computed for the visible shadow edge component. Then G_{med} is the median intensity of the ground region, and G_{adev} is the average deviation of the intensity values.

$$G_{adev} = \frac{\sum_{i=1}^{l} |Intensity(g_i) - G_{med}|}{l}.$$

  Likewise, U_{med} and U_{adev} are the median and average deviation intensity values of the shadow region, U. Then, u is defined to be the number of points in the shadow region that are darker than a threshold representing the expected ground intensity. The expected ground intensity is defined to be G_{med}. However, to account for the variations in the ground intensity values, the threshold is lowered by G_{adev}, so that a pixel whose intensity value is within G_{adev} of G_{med} is not labelled as a shadow pixel. We define the shadow color component,

$$SC = \frac{u}{|U|} - w_{sc}\frac{U_{adev}}{256}.$$

  The shadow is assumed to be uniformly dark and $\frac{U_{adev}}{256}$, the normalized average deviation of the shadow region, is used to penalize a hypothesized shadow region with non-uniform intensity. 256 is the largest possible deviation for an 8 bit image. The uniformity of the shadow region is weighted by w_{sc}, which is currently set to 0.5. In SITECITY, T is currently set to a distance of 4 pixels.

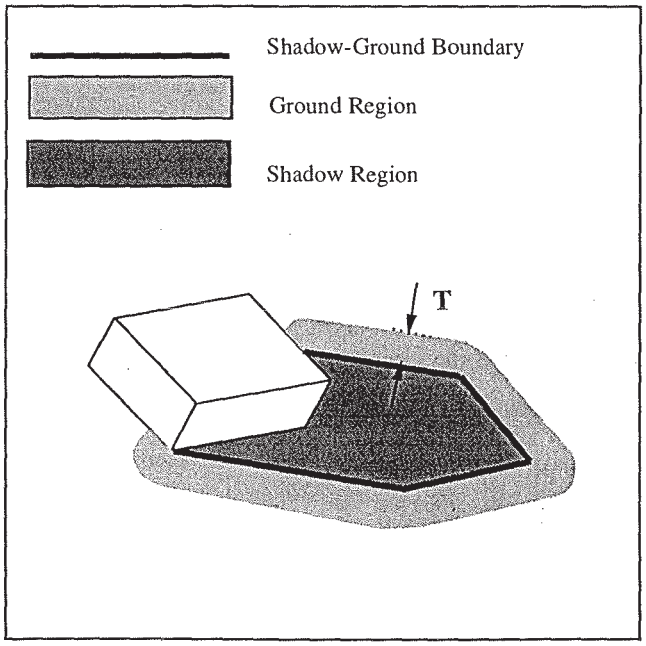


Figure 6: Illustration of shadow color component

The final verification confidence of a building model, VC, is simply:

$$VC(A, B, S, I, v) = w_1 VE(A, B, v) + w_2 EG(A, I) + w_3 VSE(S, B, v) + w_4 SC.$$

w_i are the weights for each component, and they are currently each set to a value of 1 in SITECITY. Hence, VC returns a value between 0 and 4, where 4 is the ideal confidence of a model building. This verification component is currently used for all automated processes described in this paper.

3.1.2 Edge Estimation Component

Often, it is desirable to determine the terminating endpoint of an edge of known directed orientation θ and length q in an image; the edge estimation component is used to achieve this task. Figure 7 illustrates this task. The edge estimation component generates a set of edge hypotheses such that one terminating endpoint of the edge hypothesis lies on the edge L. θ, q and the endpoints of L are inputs to the edge estimation component.

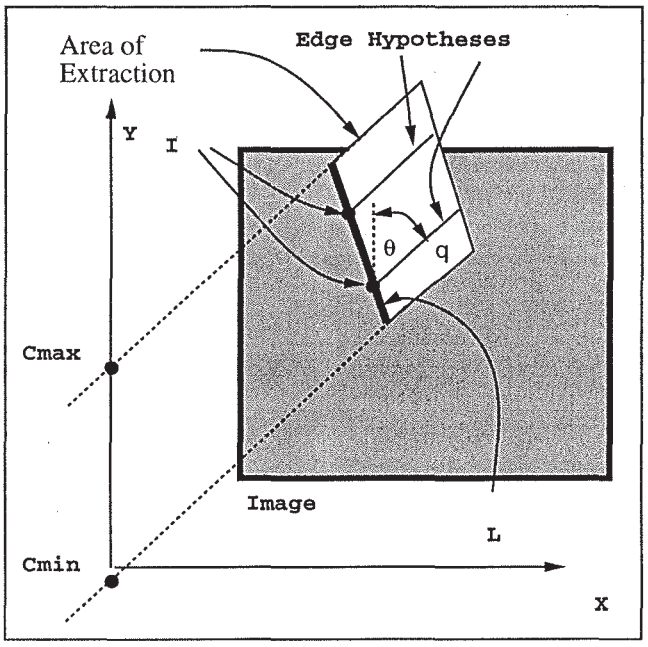


Figure 7: Edge estimation component

The line equation for one edge hypothesis is $aX + bY + c = 0$, where a and b are known via θ. Let $C = \{c_1, c_2, \ldots, c_n\}$ be the set of possible c values; then the Hough transform [4] is used to collect evidence for each c_i using image edge points extracted from the image. Clearly, the image edge points need not be extracted from the entire image. The portion of the image that needs to be considered is the parallelogram defined by L, θ and q. The parallelogram can be computed by extruding edge L in the direction of θ with a distance of q. If the number of image edge points on an edge hypothesis is less than 2, then the edge hypothesis is rejected. The minimum and maximum values in C are limited by the constraining edge, L. Let I be the set of possible edge locations for one terminating endpoint. Then I is the set of intersections of L and $aX + bY + c_i = 0$, $\forall c_i \in C$.

3.1.3 Object Matching Component

The object matching process is used to generate hypotheses for a known object in the image given a search region. The search region is defined by the automated processes that use this component. Given a projection of a 3D object to the image, the goal is to find instances of the projection of this object in the image, similar to the 2D object matching problem described in [27; 8; 46].

The brute force approach is simply to instantiate the model at every possible location within the search region and find good matches. This approach ensures us that the correct location will be in the set of initial hypotheses assuming the search region contains it, but the cost of searching can be prohibitive. Therefore, a simple heuristic is used to reduce the search space. Given the projection of a 3D object to an image, the size and the orientation of visible corners can be predicted. For each predicted model corner, all the corners within the search space that have the same orientation and size are used to generate a list of plausible hypotheses. For an image corner that has the same shape and orientation as the model corner, an object hypothesis is created by copying and positioning the the model such that the model corner is located at the position of the image corner. Object hypotheses that extend outside the search region are ignored. This set of plausible hypotheses can be pruned using the verification component. The pruning criteria are specified by the automated processes that use this component.

3.2 Automated Processes in SiteCity

The three components described in the previous section can be composed in a variety of ways to perform different tasks. The verification component is the heart of the automated system, since it defines good and bad hypotheses. Edge estimation allows us to generate hypotheses from partially specified objects, since image edges can be used to determine object extent. The object matching component can be utilized to match objects in a single image.

3.2.1 Estimate Peak Edge of a Peak Roof Building

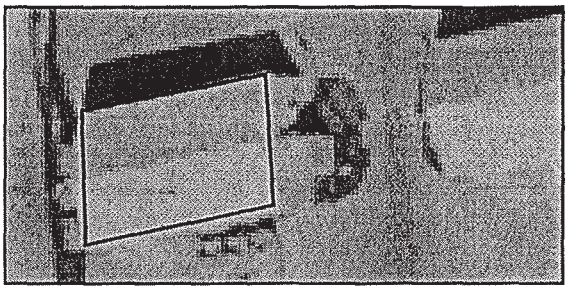
For a peak roof building model, once the user measures the rectangular outline for the roof, the first task is to estimate the position of the peak edge. According to the peak roof building model, the peak edge must be horizontal, above the rectangular roof outline, and lie halfway between two walls. Therefore, the end points of a peak edge must be on the vertical lines intersecting the roof edges perpendicular to the peak edge. These two vertical lines are the constraining edges used to define the search space in the edge estimation component. Currently, the maximum possible height (length of L in Figure 7) of the peak edge is computed using a maximum peak angle of 60 degrees. To evaluate the edge hypotheses, the verification component is used. However, the shadow components of the verification process are not used because we cannot predict the location of the peak edge shadow, without first knowing the floor location.

Every peak edge hypothesis can be exhaustively combined with every floor hypotheses to generate a building hypothesis, and each of these could then be evaluated. However, this is a time consuming process and not implemented in the current system. The peak edge hypothesis with the best confidence value returned by the verification component without shadow analysis is selected.

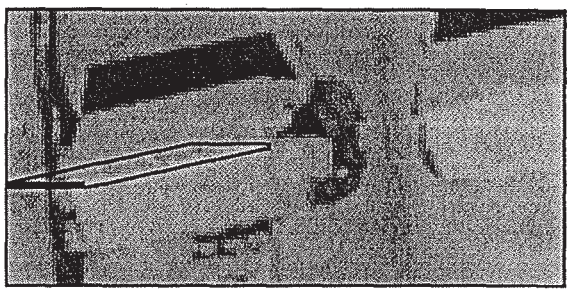
Figure 8(a) shows the user delineated building roof. The search bound for the peak and the vertical vanishing lines based on the roof are shown in Figure 8(b). Figure 8(c) shows the set of possible peak edges extracted from the image, in black. The set of possible peak edge positions, where a peak edge hypothesis intersects the vertical vanishing line, are shown as white dots.

3.2.2 Estimate Building Height

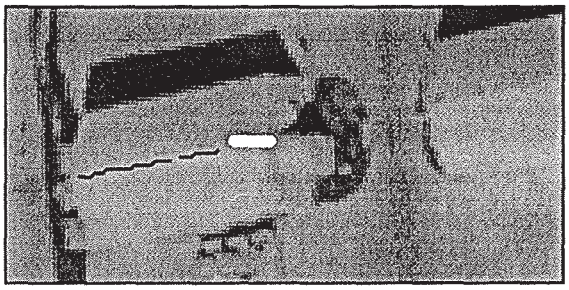
In [37], the height of a building is measured by using an imperfect sequence finding technique [1] to locate vertical edges. Another clue in the image that allows us to determine the building height is the location of the visible floor edges, where the walls of the building meet the ground. Since the delineation of the roof is given, the orientation and size of the visible floor edges can be predicted. For the defined building models in SiteCity, the floor is simply a vertical extrusion of the delineated roof outline. Given these floor edges in image space and the vanishing point geometry, we can use the edge estimation component to hypothesize a set of building hypotheses. The plausible edges are those that are within the search region bounded by the vertical vanishing line. The search length of the vertical vanishing line is defined by the roof points and the maximum height parameter set by the user. Each floor edge hypothesis produces a hypothesized building model. All the building models are viewed to be in competition with each other, because there should be only one building model for a given roof outline; therefore, the verification component is used to select the best model.



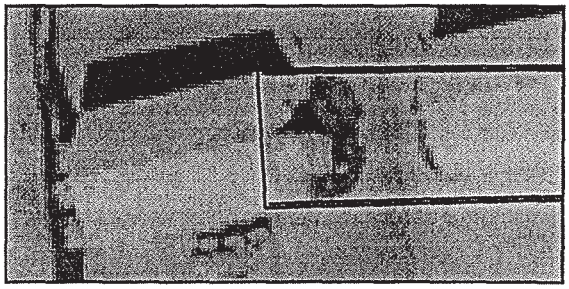
(a) Roof outline



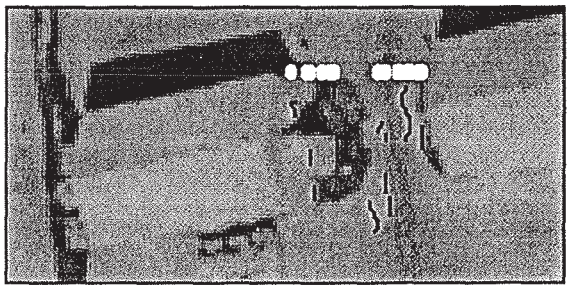
(b) Search bound for peak edge



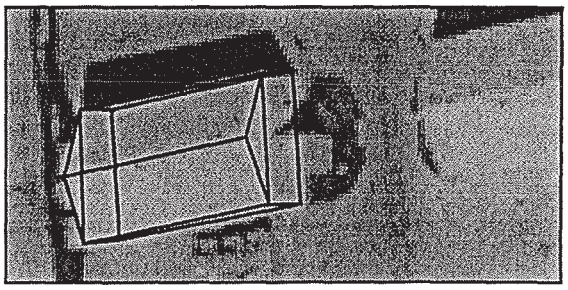
(c) Detected peak edge and possible peak edge location



(d) Vertical search bound



(e) Detected floor edges and possible floor location



(f) Detected building

Figure 8: Example of building height estimation

Figure 8(a) shows the user delineated building roof. The search bound defined by the maximum height and the vertical vanishing line based on the roof is shown in Figure 8(d). Figure 8(e) shows the set of possible floor edges extracted from the image in black. The set of possible floor corner positions shown as white dots and the building selected using the verification component are shown in Figure 8(f).

The position of the floor points for the model is used to determine the position of the building in 3D space using a single image. Since the building is assumed to be on the ground, the DEM is used to determine the elevation of the floor position. The length of the vertical edge in the image space is converted to height in object space [37], and the resulting 3D model can be projected to other images.

3.2.3 Epipolar Matching

Due to errors in the DEM, and the uncertainties of the camera parameters, the projection of a 3D object onto another image might not be correct. To achieve a good 3D measurement, it is often necessary to locate the projected object in several images.

If the location of the object is known in one image, then the search space in another image is defined by the epipolar line and the precision of the epipolar line. To account for uncertainty of the epipolar line, the epipolar line used to construct the search region is assumed to have a width. This width is the standard deviation of the epipolar line position, which is derived from the camera models. The search region is also constrained by the minimum and maximum elevation range set by the user. Given the object model in the image and the search space, the object matching component is used to generate a set of hypotheses. The hypotheses generated by the object matching component have one more constraint. All the corner points of an object hypothesis must fall within the neighborhood of their corresponding epipolar lines. The verification component is used to select the best hypothesis.

Figure 9(a) shows the initial projection of an object in an image. Figure 9(b) shows the epipolar lines associated with each vertex of the projected building object. For each epipolar line, the errors are estimated based on imaging parameters. In Figure 9(b), none of the epipolar lines go through their corresponding points in the image due to inaccuracies in imaging parameters. If these errors of the epipolar lines are not considered, it will be difficult to determine a valid search region. The search region is constructed using the epipolar lines and the error bounds. A set of plausible corners are shown in Figure 9(c). Finally, the selected building location is shown in Figure 9(d).

3.2.4 Automatic Copying

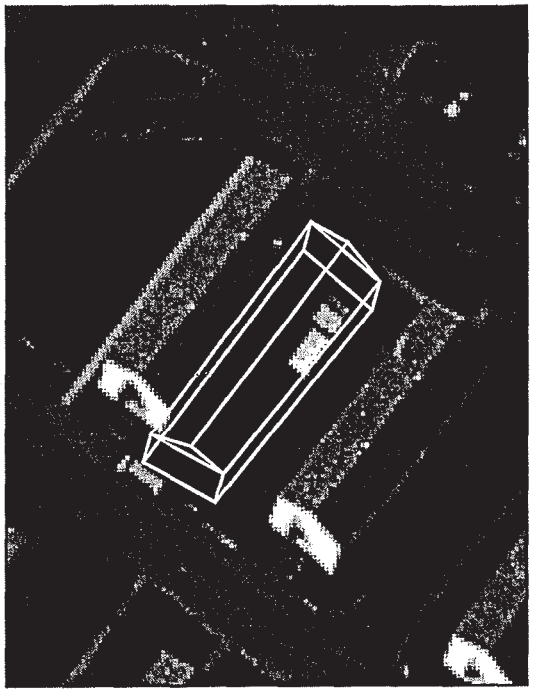
Frequently, many objects in the scene might be identical. Suburban scenes often have repetitive instances of the same buildings. Two objects are identical if they share the same shape, size and orientation. The object matching component can be used to reduce the effort of copying an object to a different part of the image. Since the object matching component requires the determination of the search area and the object, we devised three methods to specify the search region:

- **Point Cue**

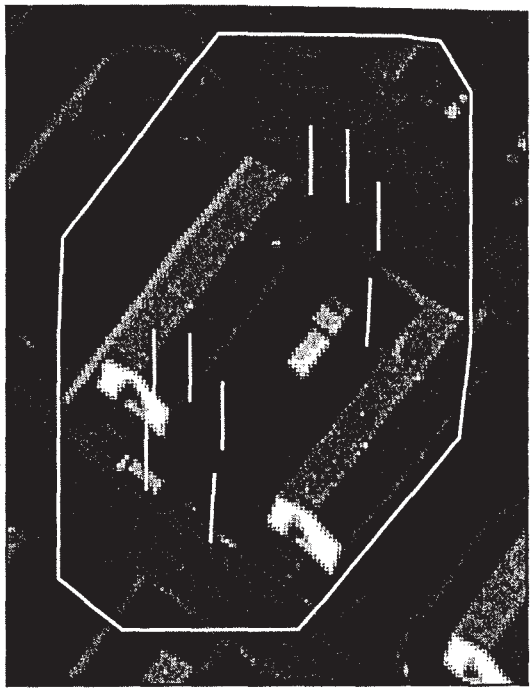
  When the goal is to duplicate only one object, a point cue can be used. To specify a point cue, a point is specified on top of the desired object as it appears on the image. The only requirement for the point cue is that it must reside within the boundary of the desired object. Figure 10(b) shows an example; the position of the point cue is specified in black and the computed search region using the point cue and the known object is shown in white. Only one object can be created from automated copy process using the point cue. The search region can be computed by finding all possible locations of the object in the image such that the object overlaps the location of the point cue. The search region is simply the union of all the possible locations.

- **Line Cue**

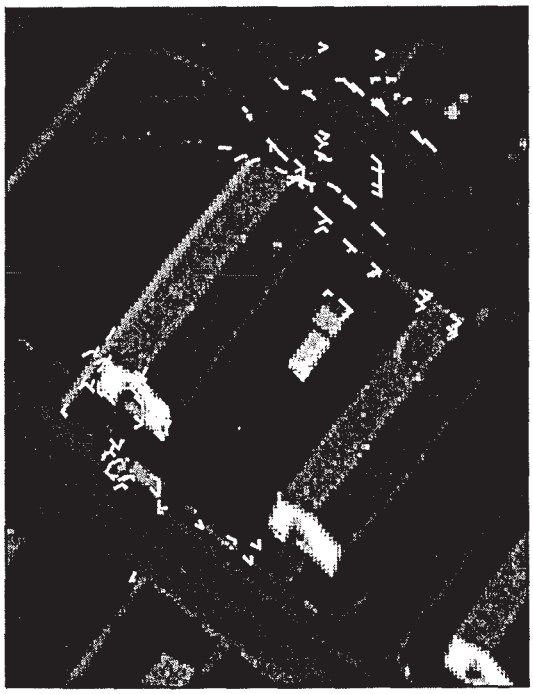
  When multiple identical objects are aligned along a line, a line cue can be used. A line cue can be specified by specifying two points on the image, and is an edge. The semantics of the line cue are that the desired objects must intersect the line. Figure 10(c) shows the line cue specified by the user in black; the computed search region using the line cue and the known object is shown in white. More



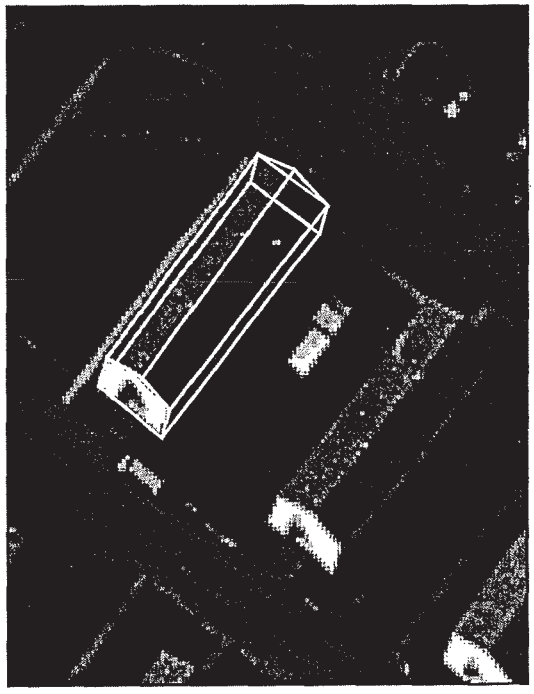
(a) Initial projection



(b) Epipolar line and search region

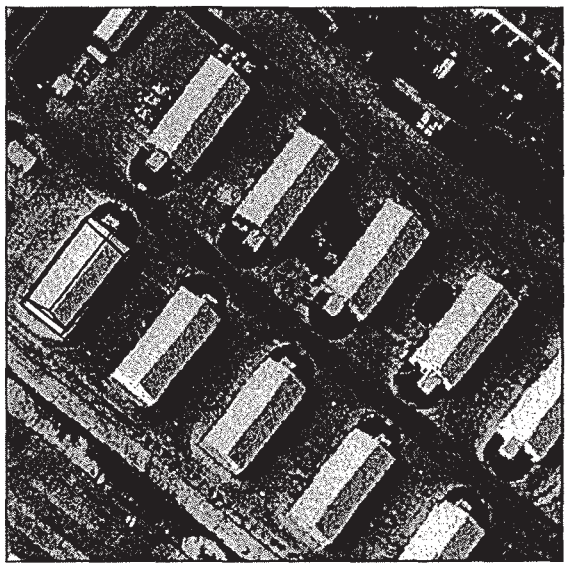


(c) Detected corners

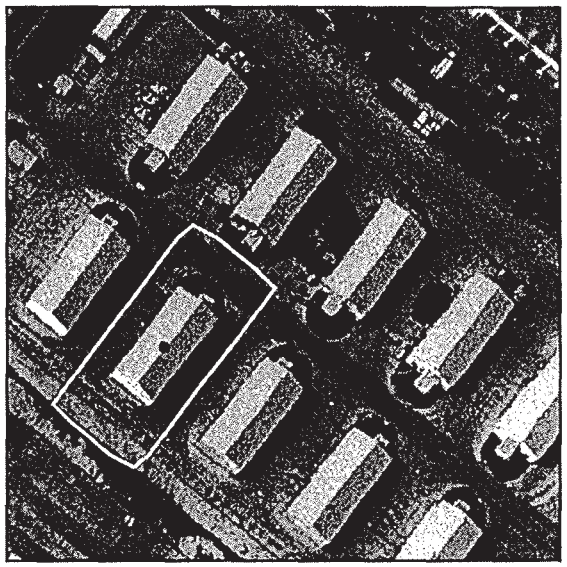


(d) Localized building

Figure 9: Example of epipolar matching

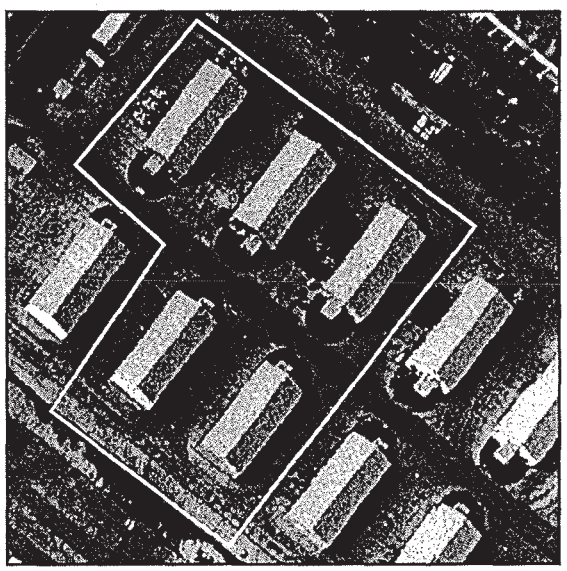


(a) Object to be copied



(b) Point cue

(c) Line cue



(d) Area cue

Figure 10: Example of cues

than one object can be created from the automated copy process using the line cue. The search region for the line cue can be computed in a similar fashion as the point cue.

- **Area Cue**

  Finally, a polygon can be drawn that serves as the search space. Figure 10(d) shows an example of the polygon cue. The desired objects must lie wholly within the polygon. The polygon cue must have at least three points.

After processing by the object matching component, a number of hypothesized object locations are determined. Most of these hypotheses are erroneous and the verification component is used to compute the confidence of each hypothesis. With a point cue, the semantics of the cue specifies the existence of only one object within the search region; therefore, the hypothesis with the best confidence is selected. For the line and area cues, the actual number of objects in the image is unknown. To reduce the number of false hypotheses, overlapping hypotheses are pruned. The best hypothesis is used to remove all other hypotheses that overlap it. When no hypothesis overlaps the best hypothesis, the second best hypothesis is used to remove all the remaining hypotheses. The process continues until all remaining hypotheses are non-overlapping. Two hypotheses overlap if their floor footprints intersect in object space.

At this stage, one possibility is to use a threshold to remove non-plausible hypotheses. Based on experiments, we found that a threshold of 2.0 prunes away the majority of false hypotheses. Another possibility is to return all non-overlapping hypotheses. Since the automated copy AP is to be used in an interactive environment, it is possible to let users delete false hypotheses. Currently, the threshold approach is used in SITECITY.

3.3 Incorporating Automated Processes For Building Measurements

Once the user delineates the roof of a building, there are three paths in which the automated processes can be used to generate a building hypothesis.

- **Single Image Building Detection (SIBD)**

  Figure 11 shows the process flow for SIBD. In this scheme, a building object is generated and verified using a single image. The 3D position of the building is computed by epipolar matching of the building objects in image space.

  After the user measures the corner points for the roof, the floor detection process is invoked to established the structure and location of the building object in the image. The floor of the building is assumed to be on the ground. The length of the vertical edges in the image are converted to meters in object space. Since the roof position differs from the ground only in elevation, the height measurement allows us to construct the building structure in the local coordinate system. The peak edge detection process is invoked if the building is a peak roof building.

  The estimated 3D structure is projected to other images. Since the elevation of the building is estimated based on the DEM, or a preset default ground elevation, and since there are uncertainties in the camera parameters, the projected structure often does not correspond to the building object in the images. Therefore, the epipolar matching process is invoked to localize the building in these images.

  Once the building object is measured in all the images, the 3D position of the object is re-triangulated and geometric constraints on the building model are applied.

- **External Building Verification (EBV)**

  In this approach (Figure 12), a set of building hypotheses is generated using a single image. Then, all the available images are used to determine the best hypothesis.

  As usual, the roof corners for the building are measured by the user, but instead of returning the best hypothesis, the peak and floor detection processes return a set of building hypotheses. Each building hypothesis is projected to other images and epipolar matching is performed to form a set of 3D building hypotheses. The confidence for each 3D building hypothesis is the combination of all the computed confidences from the different images. The hypothesis with the highest confidence is selected.

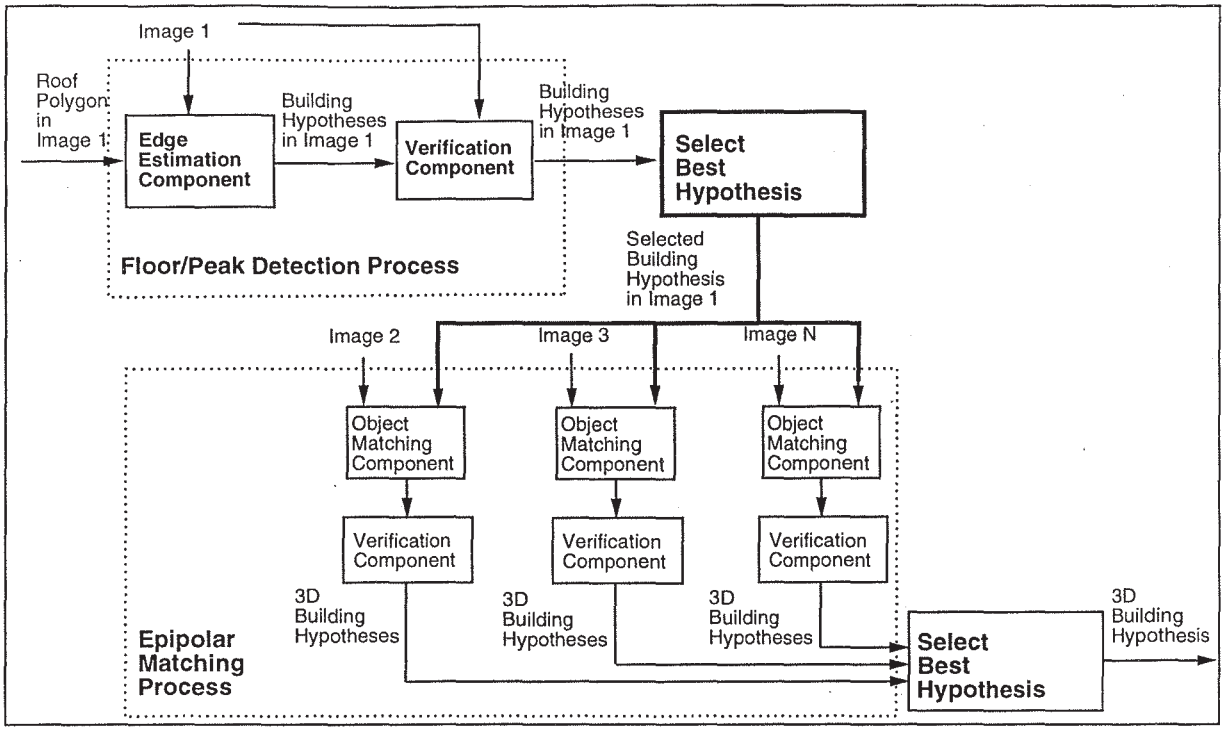


Figure 11: Process flow for single image building detection (SIBD)

- **Multi-Image Roof Matching (MIRM)**

  After the user measures a roof outline, this method takes the roof outline and performs the epipolar matching of the roof on other images and generates a set of roof hypotheses in each of the images. For each roof hypothesis in an image, peak edge and floor detection processes are used to determine a set of possible building hypotheses in that image. Three-dimensional building hypotheses are generated by triangulating building hypotheses from at least two images. All possible permutations for combining building hypotheses in image space need to be considered. The confidence for a 3D building hypothesis is the combination of the computed confidences from the building hypotheses in image space. The hypothesis with the highest confidence is selected. This method is shown in Figure 13.

The differences between the three methods can be summarized as follows: SIBD generates and verifies building hypotheses using a single image, similar to [37]. EBV generates building hypotheses using a single image, and verifies them using multiple images. MIRM generates and verifies buildings hypotheses using multiple images similar to the MultiView system described in [51]. In MultiView, the primitive object used in the matching process is corners, while MIRM matches roof delineations (polygons) in multiple images. After the corner matching process, MultiView attempts to form three dimensional polygonal descriptions by hypothesizing and linking edges with corners. In MIRM, the three-dimensional polygons are used to hypothesize three dimensional building objects.

It should be clear that SIBD has the lowest complexity in terms of combinatorics, and EBV and MIRM are more complex. Assume that for each roof outline, the peak edge and floor detection processes generate a maximum of α building hypotheses. For each hypothesis, the epipolar matching process generates a maximum of β hypotheses for each additional image that needs to be verified. Let η be the number of images used. Using SIBD, a total of $\alpha + \beta(\eta - 1)$ 3D building hypotheses can be generated. Using EBV, a total of $\alpha\beta(\eta - 1)$ 3D building hypotheses are produced because for a building hypothesis in the image where it was hypothesized, β building hypotheses are generated using the epipolar matching process for the other

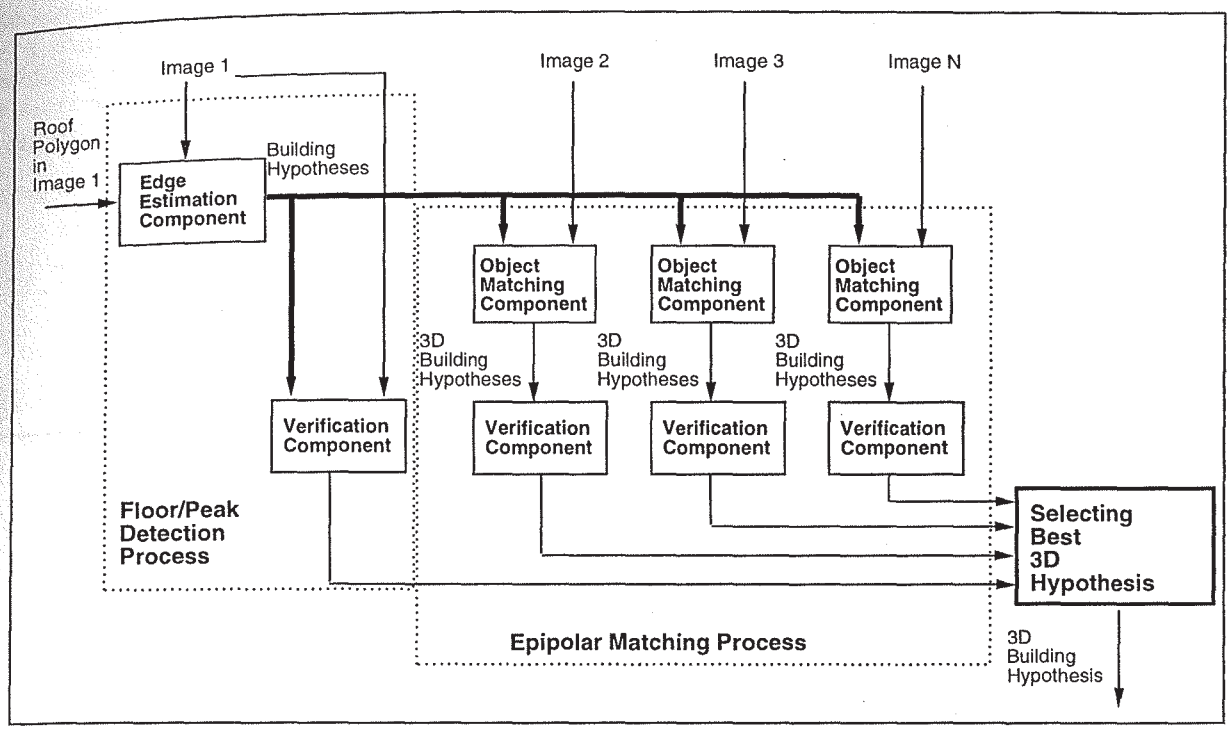


Figure 12: Process flow for external building verification (EBV)

$(\eta - 1)$ images, producing $\alpha\beta$ 3D building hypotheses for each of these images. For MIRM, β roof hypotheses can be generated from a single roof outline for each additional image. Each roof hypothesis can generate another α building hypotheses, resulting in $\alpha\beta$ image space building hypotheses for each additional image. Combining image space building hypotheses in all possible ways gives $\sum_{i=2}^{\eta-1} \binom{\eta-1}{i} (\alpha\beta)^i$ 3D building hypotheses that need to be evaluated and is exponential in the number of images in the worst case.

The preceding analysis considers the worst case scenario, as the values of α and β vary across images. Nonetheless, our experiments show that both MIRM and EBV are currently slow due to combinatorics, and for reasons of speed, only SIBD is currently used within SITECITY. The main drawback of using SIBD is that the user needs to select the images with care, because it is not always possible to extract heights and find floors in vertical images, and often, walls and floors might be occluded in one view and not in another.

3.4 Measurement Errors

Due to uncertainty in the imaging parameters and the image measurements, the calculated 3D position of the measured points will also be uncertain. In order to assess the quality of these measurements, a least-squares optimization with error propagation [41] is implemented to quantify the precision of all 3D measurements. The least-squares approach requires as input an estimate of the initial measurement precision.

There have been numerous attempts to quantify image measurement precision [16; 23; 58; 57; 56]. These reports show that the measurement precision on a digital image depends on Signal to Noise Ratio (SNR), quantization level and pixel size of the digital image. Since it is difficult to derive the SNR for an arbitrary image, and the quantization level is typically restricted to 8 bits per pixel for grey scale images, only the effect of pixel size is considered. The size of the pixel on a monitor is affected by two factors. The first is the resolution of the scanning process, in which an image on film is discretized into a digital image. The other factor is the displayed resolution, where the pixels on the screen can be artificially enlarged, and

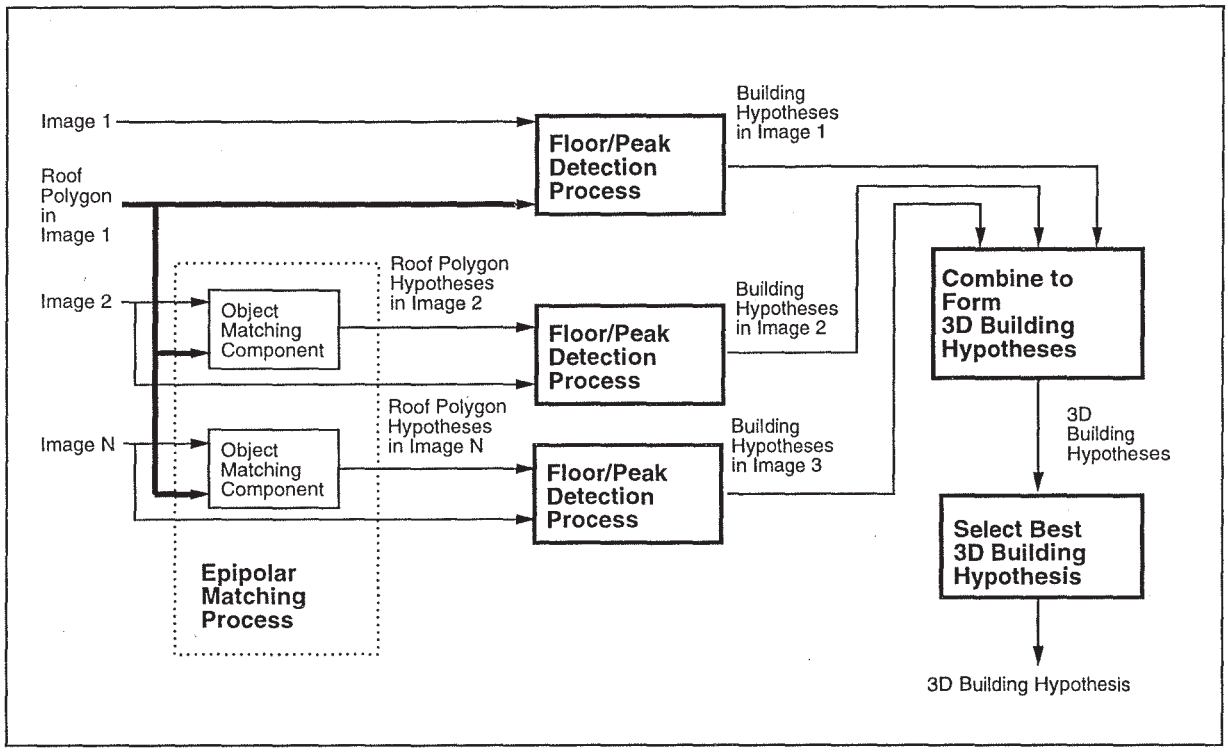


Figure 13: Process flow for multi-image roof matching (MIRM)

values between pixels are interpolated to increase the resolution of the original digital image. Since an end user often does not have control of the scanning resolution, SITECITY only considers the effect of displayed resolution on the accuracy of image measurements. The initial image measurement error for a measurement is based on the classification of the measurement. Every measurement is classified into one of the following four categories:

- **User Input**

  A measurement labeled *user input* is manually measured by users. The measurement error of this type of measurement is defined to be

  $$\frac{Image\ Sigma}{Zoom\ Level}.$$

  Image Sigma is a measure of the estimated human measurement error in terms of pixels at a zoom level of 1 and is a user adjustable parameter. *Zoom Level* is a measure of the display resolution in which a measurement is performed; more details can be observed at higher zoom level. At lower resolutions (lower zoom level), the expected precision of user measurements is lower than at higher resolutions.

- **Automated Process**

  This measurement was performed by an automated process and the measurement error depends on the accuracy of the process. If the process performs on a sub-pixel level, the measurement error can be less than 1 pixel. The measurement error of automated processes also depends on the quantization level and SNR of an image. The measurement error denotes the geometric accuracy of the measurement, not the correctness or confidence of that measurement.

- **Initial Position**

  Sometime, SITECITY has to set the position of a required measurement at some initial position, and when it occurs, the measurement is classified as *Initial Position.* For example, when the users copies a point without specifying its location, SITECITY has to display the point somewhere on the image. If there is no information about the nature of the point, it will set the position of the point to some default location and set the measurement error to some appropriately large value. In the current system, a measurement error of 100 pixels is used. This is required to estimate the uncertainties of the triangulated 3D objects; a 3D object whose image measurements have larger measurement error should be more uncertain.

- **Invisible**

  In some cases, a point on a 3D object might be occluded in some images; the location of the point is inferred and cannot be directly measured. The measurement error of such a point need not be larger than the dimensions of the occluding region because if the measurement error is larger than the occluding region, then it cannot be occluded by the occluding region. In SITECITY, the image sigma for the invisible point is defined to be the greatest distance of the point from any point on the occluding region. Let P be the invisible point, Q be the occluding region and $Q = \{q_1, q_2, \ldots, q_n\}$ is a finite set of points. Then the measurement error for P is

  $$MAX(\|P - q_k\|), \forall q_k \in Q.$$

Every measured image point can be assigned one of the above four classifications. The image sigma for each point, derived from these classifications, is used as an initial error estimate for the least-squares error propagation to assess the precision of the triangulated 3D measurement. Table 1 shows an example of measurement classifications using the manually measured building in Figure 14. The building consists of eight points and was manually measured on two images. The standard deviation matrices for each triangulated three-dimensional point are also shown to illustrate the effect of the image sigma on the precision of the 3D point. Point 3 is invisible in both images, hence, the triangulated 3D point for point 3 has poor precision and its location in object space appears to be inaccurate. The triangulated building is shown in Figure 14(c), and it does not agree with our expectation of the shape of the building. Therefore, geometric constraints are used to apply knowledge about the shape of the model in the triangulation process.

3.5 Geometric Constraints

Often, despite careful image measurements, the three dimensional object calculated by triangulating multiple image measurements does not conform to our precise expectations for the real 3D object. In addition, direct image measurements are sometimes impossible due to occlusions or shadows. Therefore, geometric constraints [35] that incorporate knowledge about the object shape are utilized to produce a more accurate 3D model.

Figure 14 shows an example of a rectangular building object measured in three images. Even though the measured building points in the images are perceived to be correct, the triangulated three dimensional building does not have the expected shape. One of the floor corner points, point 3, has significant error, since it is not visible in any of the images. As expected, the covariance of that point is relatively large compared to other visible points (See Table 1). Figure 14(d) shows the 3D building model after the application of geometric constraints, and the precision for the constrained model is also shown in Table 1. These types of geometric constraints are *intrinsic constraints*. Their purpose is to constrain the measured object to the geometric model of the object. The set of *intrinsic* geometric constraints for each building type is different, and they are derived from the definition of the building model. For example, a flat roof building model has the following *intrinsic constraints*:

1. All the points on the roof must be coplanar and horizontal.
2. All the points on the floor must also be coplanar and horizontal.
3. Every *vertical* line on the wall must be vertical.

Another application of geometric constraints is the construction of complex objects by composition of a few primitive object types; this alleviates the need to define a model for every type of object one can encounter. These types of constraints are called *external constraints*; examples are shown in Figures 15 and 16. The complex building is composed of 4 sections. Two smaller rectilinear buildings, A and B, sit on top of the main rectilinear building, C and a smaller rectilinear building, D, is adjacent to building C. Without geometric constraints, the measured building can have gaps or intersections and its components may not be properly aligned. The geometric constraints are specified (Figure 15(d)) such that the floors of building A and B are coplanar with the roof of building C. The adjacent walls and the connecting walls between buildings C and D are also coplanar. Figure 16 shows the 3D building before and after the application of geometric constraints.

3.6 System Parameters

One requirement for the automated processes according to the criteria described in Section 2.2 is to have as few parameters as possible. Due to the nature of the problems, it is difficult to design image understanding algorithms to be free of parameters. Therefore, we have decided to hide most of the unavoidable parameters from the user. These hidden parameters are hardwired into SITECITY and are defined throughout the descriptions of the automated processes and their components. Only four parameters are alterable by users.

- **Image Sigma**

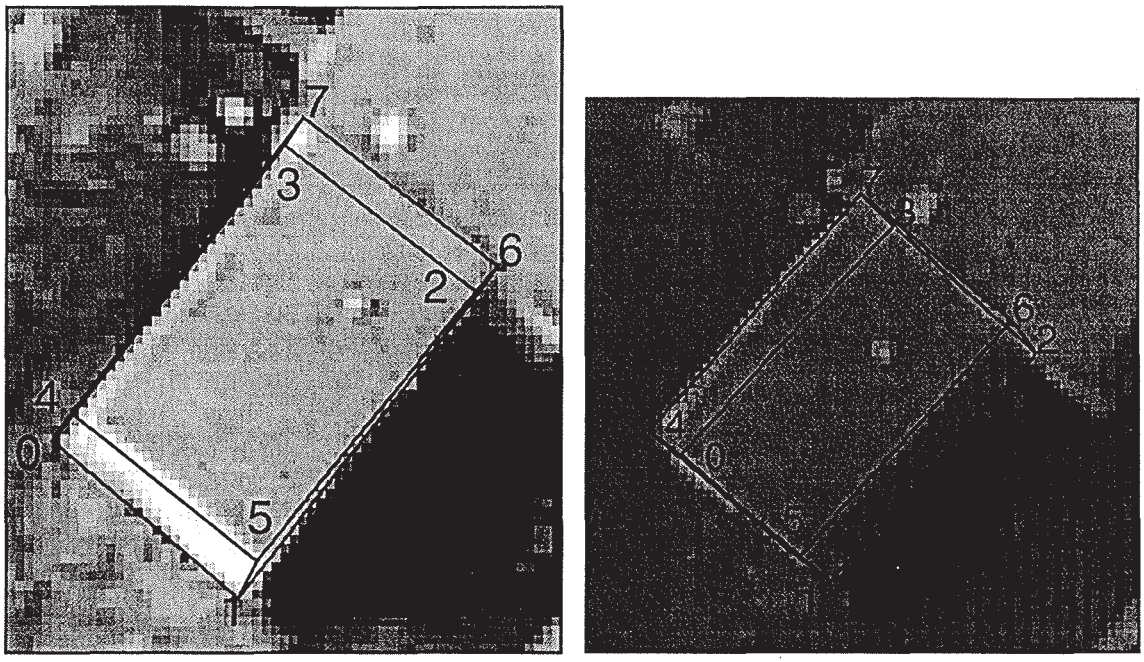
  This parameter determines the expected error in a user measurement on the image at its original resolution (See Section 3.4). The units for *image sigma* is pixels.

- **Error Propagation Sensitivity**

  Error propagation sensitivity defines the trustworthiness of the uncertainty estimate based on camera parameters. This error is used to construct search regions in the automated processes. Ideally, the error bound should be as small as possible while still remaining valid for the entire image. The estimated error is multiplied by the error propagation sensitivity to produce the error value used by the automated processes. If the error propagation sensitivity is greater than 1.0, then SITECITY treats the results of error propagation as underestimates of the true error. If this value is less than 1.0, the system treats the error propagation results as overestimates of the true error.

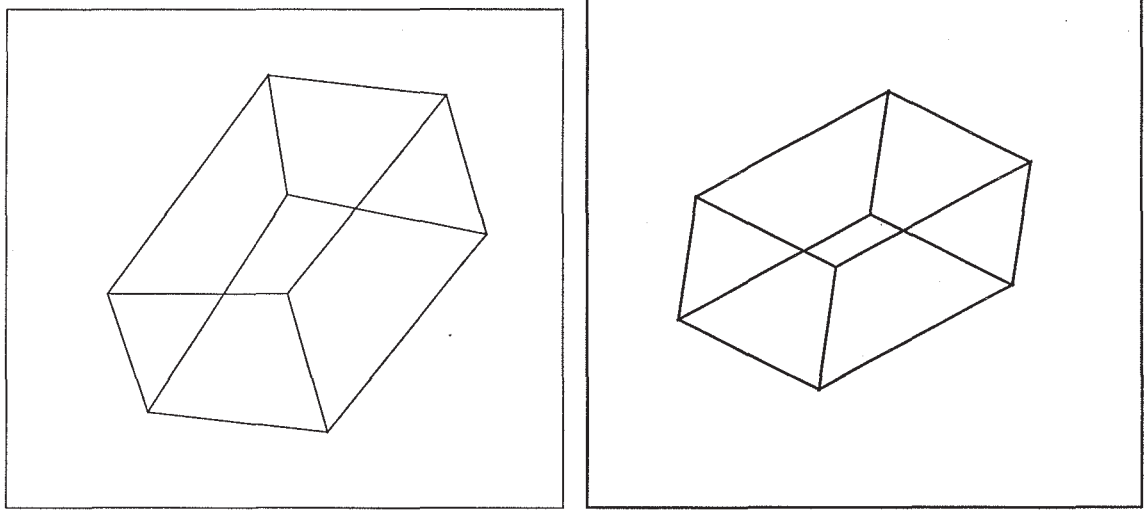
Point ID	fhrad1 Classification	fhrad3 Classification	triangulated 3D standard deviation matrix	constrained 3D standard deviation matrix
0	user input	invisible	3.00 0.35 0.57 0.35 2.70 0.57 0.57 0.57 8.50	1.80 0.16 0.29 0.16 1.70 0.28 0.29 0.28 4.00
1	user input	user input	3.00 0.34 0.57 0.34 2.70 0.56 0.57 0.56 8.50	1.80 0.14 0.28 0.14 1.70 0.27 0.28 0.27 4.00
2	invisible	user input	3.10 0.35 0.59 0.35 2.70 0.55 0.59 0.55 8.50	1.90 0.15 0.30 0.15 1.70 0.27 0.30 0.27 4.00
3	invisible	invisible	5.80 0.39 0.89 0.39 2.80 0.48 0.89 0.48 18.00	1.90 0.08 0.36 0.08 1.70 0.17 0.36 0.17 4.00
4	user input	user input	3.00 0.35 0.57 0.35 2.70 0.57 0.57 0.57 8.40	1.80 0.15 0.28 0.15 1.70 0.28 0.28 0.28 3.70
5	user input	user input	3.00 0.34 0.57 0.34 2.70 0.56 0.57 0.56 8.40	1.80 0.14 0.27 0.14 1.70 0.27 0.27 0.27 3.70
6	user input	user input	3.00 0.35 0.59 0.35 2.70 0.55 0.59 0.55 8.40	1.90 0.15 0.29 0.15 1.70 0.26 0.29 0.26 3.70
7	user input	user input	3.00 0.36 0.59 0.36 2.70 0.57 0.59 0.57 8.40	1.90 0.08 0.30 0.08 1.70 0.24 0.30 0.24 3.70

Table 1: Example of measurement errors



(a) Measurements in fhrad1

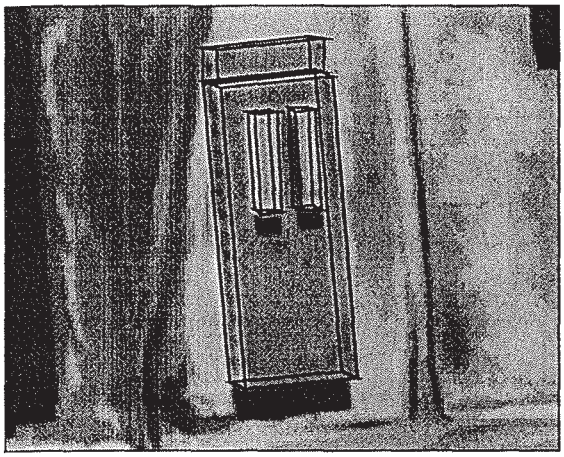
(b) Measurements in fhrad3



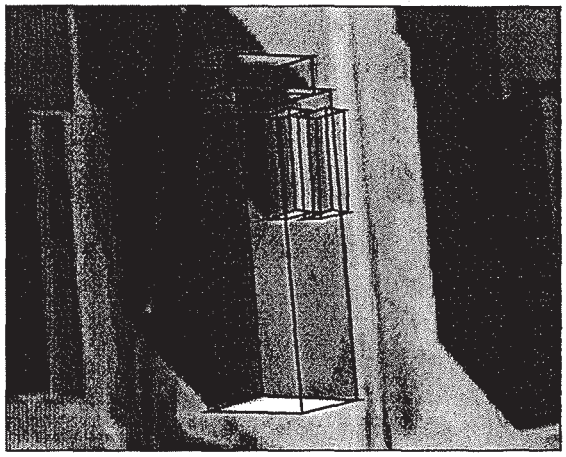
(c) Triangulated 3D building

(d) Result of applying intrinsic geometric constraints

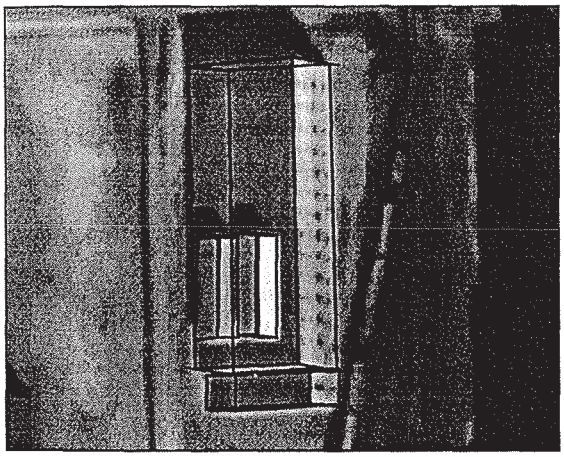
Figure 14: Example of building measurement without geometric constraints



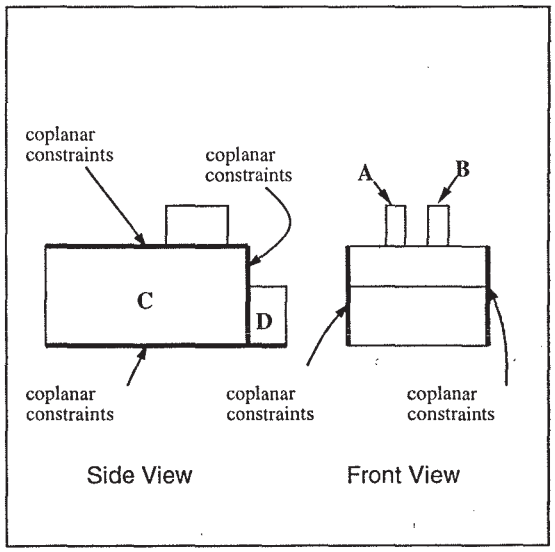
(a) Measurements in j24



(b) Measurements in j2

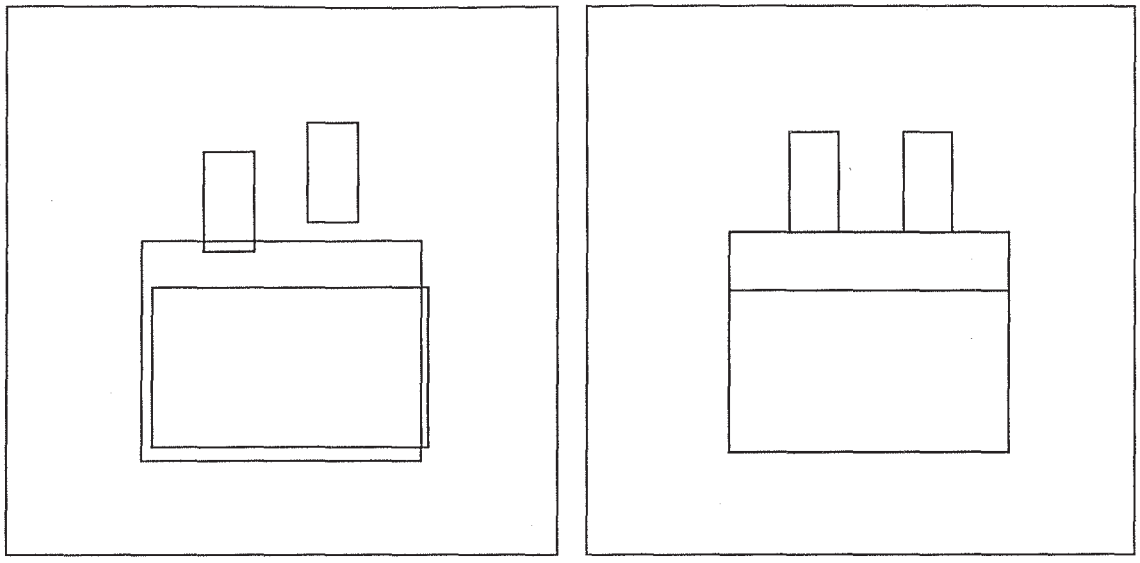


(c) Measurements in j4



(d) External geometric constraints

Figure 15: Example of complex building for external geometric constraints



(a) Before applying external geometric constraints (b) After applying external geometric constraints

Figure 16: Front view of the complex building before and after external geometric constraints

- **Minimum Elevation** and **Maximum Elevation**

  The minimum and maximum elevation define the elevation range of the scene. These two values can be interpreted in two ways. When applied to epipolar matching, these values define the range of heights in which to search for a match relative to the default ground elevation or the DEM elevation value. When applied to floor and peak detection, the difference between the maximum and minimum elevation defines the maximum building height.

3.7 Photogrammetric Tools

Three of the system parameters can be estimated using the photogrammetric tools provided in SITECITY. In addition, these photogrammetric tools can be used to *sanity check* the image orientation parameters. These photogrammetric tools operate on the photogrammetric control points, manually measured as corresponding points in each of the images. Photogrammetric control points are point objects that are used by the photogrammetric functions built into SITECITY. They are indistinguishable from point objects, except that point objects do not interact with the photogrammetric functions.

The following methods are used to *sanity check* the image orientation parameters:

- **Displaying Overlapping MBR:** This method takes the corners of the image in the window where the menu was activated and projects the corner points to other images. The elevation of the corner points are assumed to be at the ground level, defined by either the DEM or the default elevation value. This method is useful for determining image overlap for multiple images and checking the correctness of the ground elevation value.

- **Displaying Epipolar Line and Error Bound:** This method calculates and displays epipolar line and error propagation for all the measured photogrammetric control points. For each control point, the epipolar line, the zero position and a bounding control polygon representing the error of the epipolar line are computed and drawn. The zero position is the image to world projection of the control point, and represents the zero disparity location. The length of the epipolar line represents the *minimum* and

maximum elevation range. The size of the bounding polygon shows the uncertainty of the epipolar line projection; a large polygon means a large expected error in the position of the epipolar line.

- **World to Image Projection:** This method projects the triangulated 3D object to all images. The 3D objects should project to the appropriate locations in the images. The difference between the projected 3D object and the actual image object is the consequence of inaccuracies in the camera model and measurement errors.

The following two methods assist users in defining system parameters:

- **Compute Error Propagation Sensitivity:** This function uses the control points to compute the system parameter *error propagation sensitivity*. This is accomplished by adjusting the *error propagation sensitivity* value for all images such that all the control points lie within the estimated position and error bound.

- **Compute Minimum and Maximum Elevation:** The same technique can be applied to the calculation of the minimum and maximum search elevation, with respect to the ground as given by a Digital Elevation Map (DEM) or a default elevation.

3.8 Summary

A set of automated processes is designed and implemented to satisfy the criteria of automated processes defined in Section 2.2. The automated processes are modular and flexible enough to be used for a variety of applications. SITECITY only requires four intuitive parameters. Combined with the photogrammetric functions, a method is devised to estimate these parameters interactively. Since *Image sigma* estimates the accuracy of a measurement performed by a user, its value needs to be determined by individual users. Typically, the value of one pixel is used. The automated processes are designed such that they are part of the measurement processes and errors of the automated processes should not, in theory, increase the amount of work for the user. The remaining issue is the determination of the performance of the automated processes and the usability of SITECITY to test this theory.

4 Evaluation of the Semi-Automated System

A semi-automated system can be broken down into two components, the user interface aspects and the automated processes. For the purpose of this paper, we are interested in two aspects of a semi-automated system: the performance of the automated processes and the impact of the automated processes on the semi-automated system. The first issue addresses the validity of the automated processes, and the second one evaluates the usefulness of the automated processes in SITECITY. Based on these two issues, we pose the following three questions which we will attempt to answer in the following sections:

1. **Do the automated processes introduce bias into the measurement processes?**

   One concern about the semi-automated approach is that it might induce a bias in measurements. Since the purpose of the automated processes is to delineate buildings, does a user become lazy in the process and stop correcting mistakes, however minor, made by the automated processes?

   This question can be answered by comparing points measured using the fully manual approach and the semi-automated approach; this analysis is presented in section 4.4.

2. **Given partially delineated building structures, do the automated processes detect and delineate the buildings correctly?**

   In the literature for the fully automated building detection systems, there are currently two approaches for evaluating the performance of a system. The first method is the pixel and voxel based evaluation [37]. In this approach, image space or object space is discretized into pixel or voxel units. Each pixel/voxel unit is labelled either as a building or background pixel/voxel using *ground truth*. The same labelling process is performed with buildings hypothesized using an automated system. Both sets of pixel/voxel

labellings are compared to evaluate the performance of the automated system. The dimensional method described in [50] concerns the delineation of the detected building objects in object space. A 3D *ground truth* building is matched to an automatically generated 3D building hypothesis based on shape, dimension and overlap. If a hypothesized building cannot be matched to a ground truth building, it is considered to be an error. Once a building hypothesis is matched to a ground truth building, the dimensions of the building object, such as length, width and height, and the position and orientation of the building are compared between the ground truth building and the hypothesized building.

However, these two metrics are not appropriate for the evaluation of a semi-automated system. The pixel/voxel approach is concerned mainly with the detection rate, whether or not a pixel/voxel is classified correctly by a system. It does not provide a good evaluation for building delineation. In addition, since the semi-automated system is given a partially delineated building, the detection rate for the semi-automated system should be essentially good. On the other hand, the dimensional approach is performed in object space and attempts to evaluate the accuracy of 3D building hypotheses. Dimensional discrepancies in object space are a direct consequence of the measurement errors in image space and are affected by the errors in the camera models. It will be difficult to isolate errors in the automated processes from the measurement errors and errors of camera models. In addition, the automated processes in SITECITY are primarily image based algorithms. Therefore, it is preferable to perform the evaluation of building delineation in image space.

For SITECITY, another method is proposed to judge the accuracy of the building detection and delineation. In this approach, we assume that every point on a hypothesized building can be matched to every point on a ground truth building without ambiguity; i.e., there exists a one-to-one matching between all ground truth and hypothesized points. The discrepancy between the matched ground truth and hypothesized points is used to determine the accuracy of the automated building detection and delineation. The analysis of the automated processes in SITECITY for delineation of buildings is presented in Section 4.5.

3. **Is the inclusion of the automated processes helpful to users?**

   A full-fledged usability analysis for a semi-automated system is difficult and is beyond the scope of this paper, because many ergonomic and user interface issues are involved. However, assessing the effects of the automated processes in a fully manual system allows us to determine the usability of the automated processes within a system. To accomplish this analysis, twelve subjects were used to measure three scenes using both the semi-automated and the fully manual system. The fully manual system is identical to the semi-automated system, except that the automated processes in SITECITY are deactivated. By analyzing the tasks, execution, and duration of these measurements, the impact of the automated processes can be determined, and allows us to identify weaknesses of the current system. This analysis is described in Section 4.6.

To accomplish these analyses, human subjects are needed to perform measurement tasks using both the semi-automated and the fully manual system. The use of human subjects in an experiment presents numerous pitfalls that can confound the results of an experiment. Therefore, we propose the following guidelines to assist the experimental design for the evaluation of semi-automated feature extraction systems.

- *The fully manual system should be identical to the semi-automated system, except that the automated processes are deactivated.* If the fully manual system and the semi-automated system have different user interfaces, a different set of unit tasks might be performed. In this case, the performance evaluation based on the task analysis approach might not be appropriate because unit tasks from different systems might have different costs.

- A system can be biased toward a particular type of user or imagery. To reduce bias due to variations in users, *as many subjects as possible should be used in the study.* Likewise, *a variety of images covering a wide range of viewing angles and scene content should be used to avoid bias towards a particular type of imagery.*

- According to the definition of *user cost* presented in Section 2, the usability of the automated processes can be determined by analyzing the operations of users. *Therefore, the system will need to be*

instrumented to record every operation during the experiment. Ideally, every action of a user during the experiment should be recorded. By analyzing the tasks, execution, and duration of unit tasks performed during the experiment, the impact of the automated processes can be determined.

- In the feature extraction task, different users might extract different features when given the same image. This will introduce difficulties when the extracted features are to be compared. In addition, some users may spend more time to achieve better accuracy in the feature extraction tasks than other less diligent users. The *user cost* between these two users will not be comparable because results of differing quality are produced. *Therefore, the subjects will need to be instructed to perform the measurements and the extraction tasks at the same standard of performance.*

4.1 Description of the Scenes

Three scenes were selected for the experiments. One scene with a single building was designed as an example allowing a subject to familiarize himself with SITECITY. Two of the scenes are sparsely populated, with one of them having many buildings of similar shape. These scenes are identified and described below.

- **Radt9-A**

  Figure 17 shows a lone peak roof building model in four images. This scene is part of the Fort Hood aerial images distributed under the RADIUS program. It consists of two near-vertical and two oblique shots. The two oblique shots are designated with *ob* and *wob* (wide oblique) suffix. The primary use of this scene is to familiarize the subjects with SITECITY.

- **Radt5**

  Figure 18 shows the radt5 scene which is part of the four Fort Hood aerial images. There are eleven peak roof buildings and 1 rectilinear flat roof building visible in all four images. Of the 11 peak roof buildings, one row of seven buildings has the same size and another row of four buildings have shorter height.

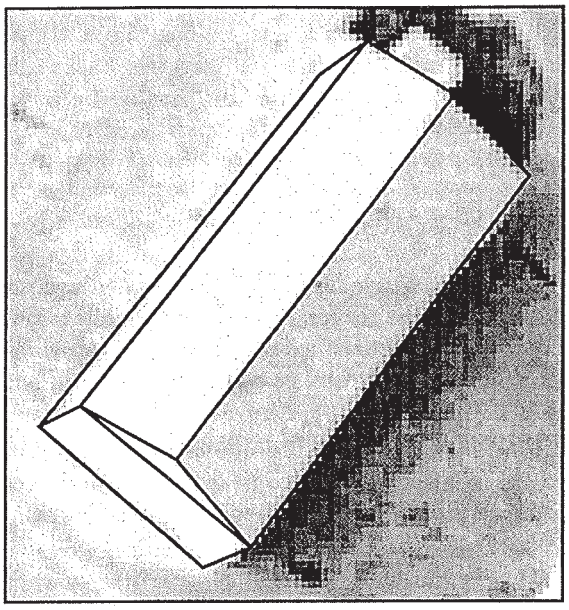
- **Radt10**

  Figure 19 shows the radt10 scene. Once again, this scene is part of the four Fort Hood aerial images. This scene consists of four rectilinear flat roof buildings and three peak roof buildings. There are inter-building occlusions in the scene, where one building partially occludes another building.

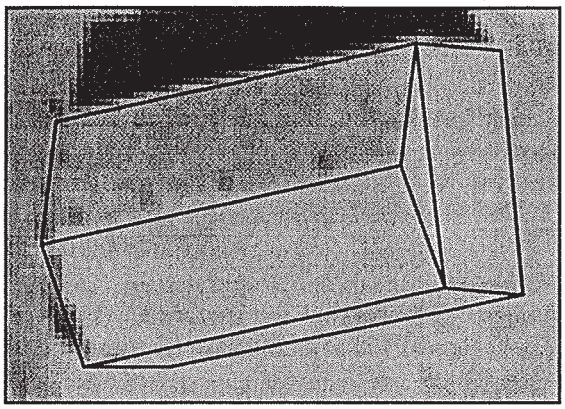
4.2 Description of the Methodology

For the experiment, twelve subjects were used. Eight of the subjects are undergraduate students working in the Digital Mapping Laboratory, and others are full-time members of the lab. Many of the subjects are familiar with the building detection tasks, however all the subjects are novices to SITECITY. For each subject, approximately half an hour was spent training them how to use SITECITY. Each subject was given the same instructions. Subjects were asked to measure buildings. A building is measured in SITECITY by measuring the vertices of the building in images. Subjects were asked to measure these vertices as accurately and as fast as possible.

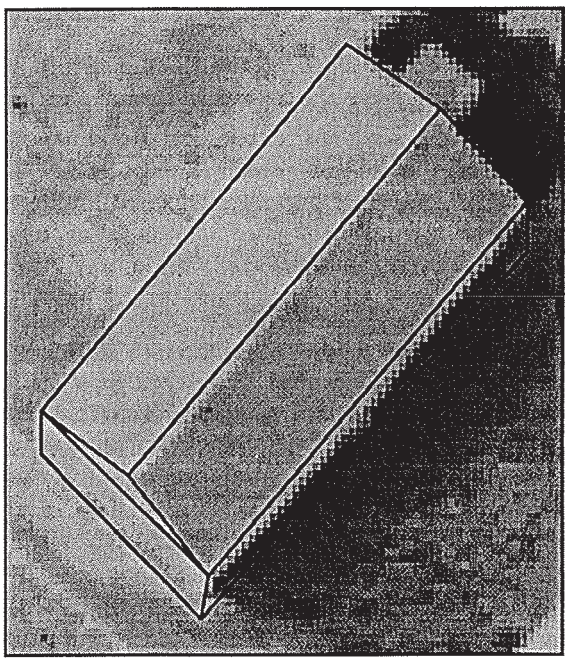
Each subject measured the buildings in radt9, radt5 and radt10 twice, with and without the automated processes. The radt9 scene is used for training purpose. In the semi-automated system, subjects do not have control over the automated processes. To reduce any bias caused by measurement order, the subjects are randomly broken into four groups, as shown in table 2. Each group contains three subjects, two of which are undergraduate students. The differences in the groups are the order in which radt5 and radt10 scenes are measured and for a given scene, whether the fully manual (m) or semi-automated (ap) approach is performed first. The reason for each subject measuring a scene using both methods is to account for variations in the users. Some users are more meticulous, while others are lazy; it is difficult to compare the performance for different subjects when they use different methods, especially when the size of the subject pool is small. We hope that by having subjects perform measurement tasks using both methods, some user variations might



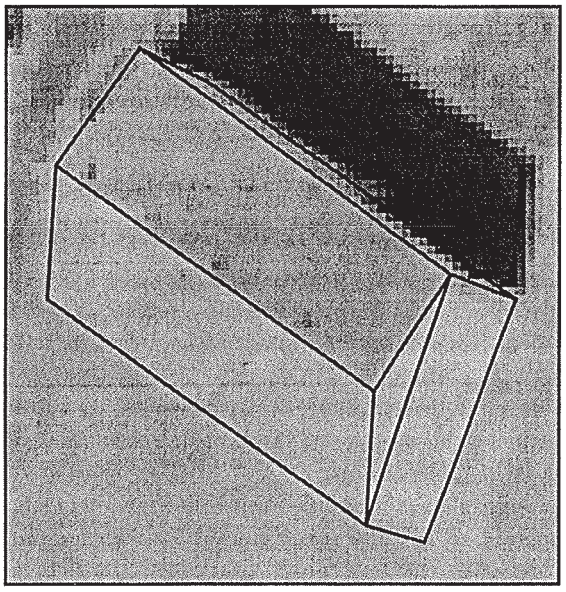
(a) Radt9-A



(b) Radt9ob-A

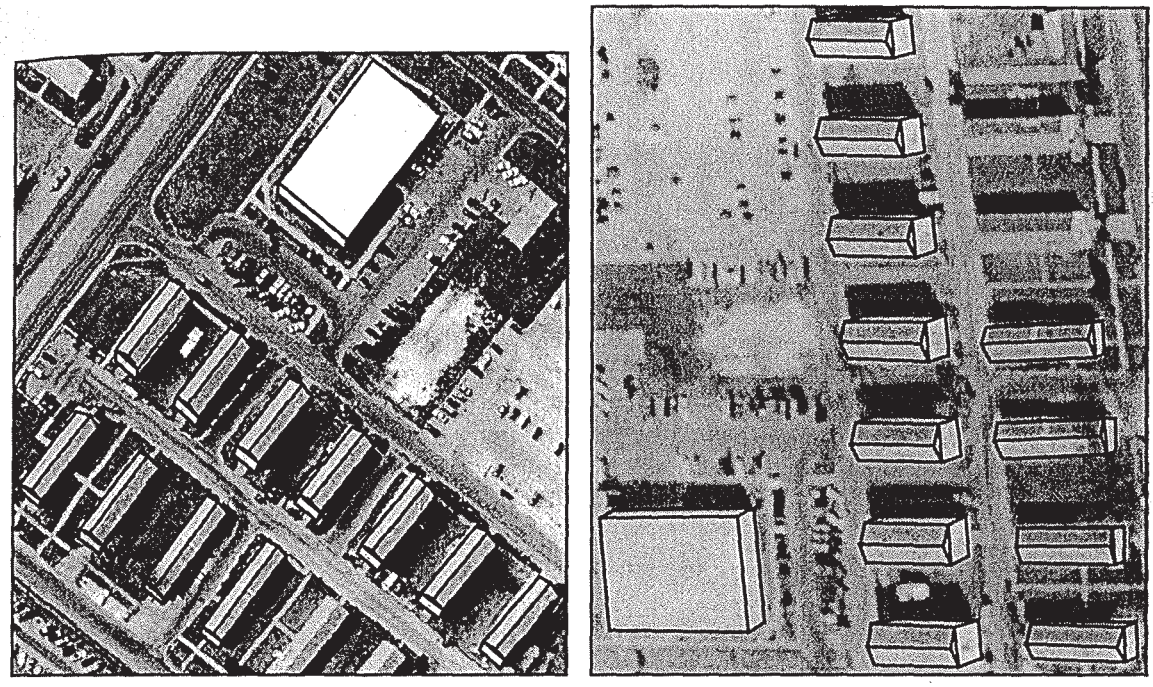


(c) Radt9s-A



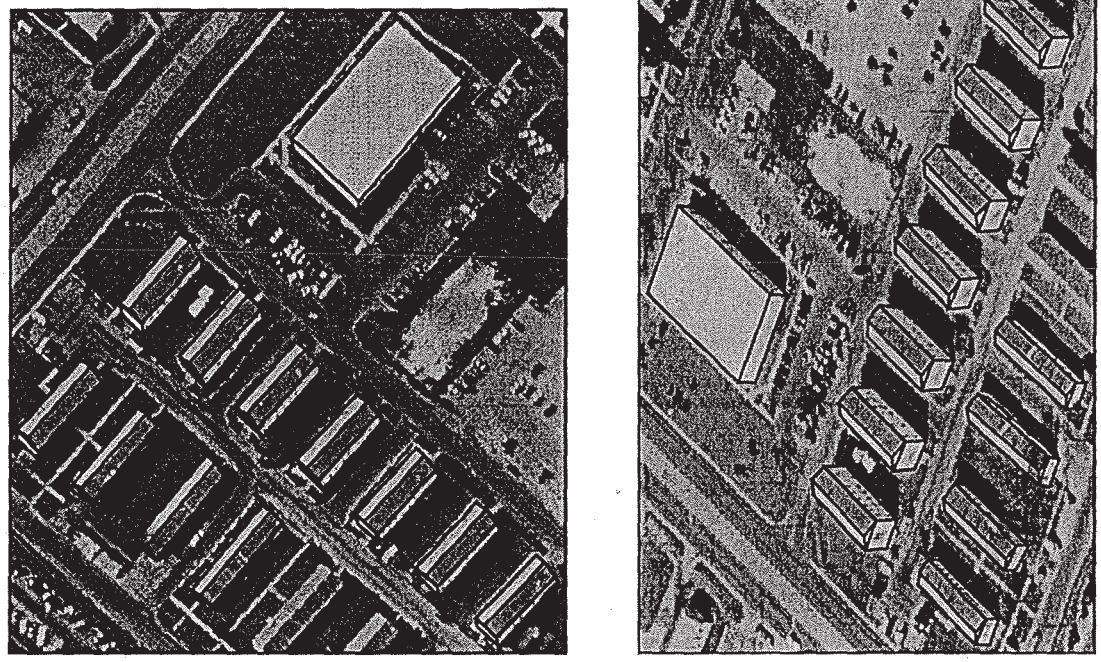
(d) Radt9wob-A

Figure 17: Scene 1 : Radt9-A



(a) Radt5

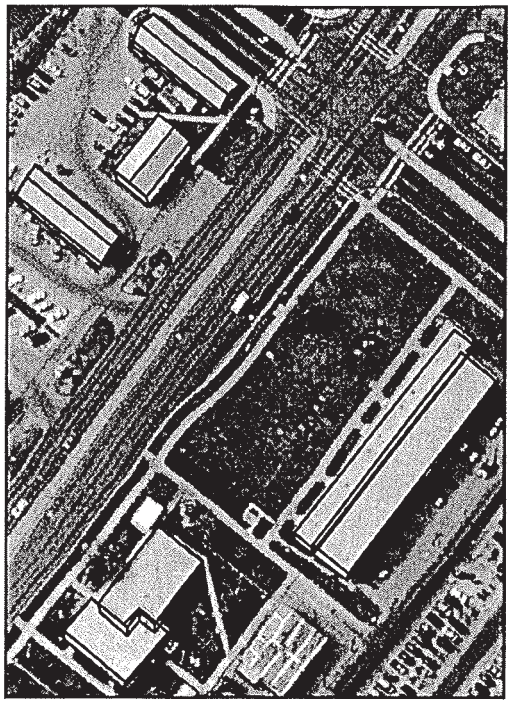
(b) Radt5ob



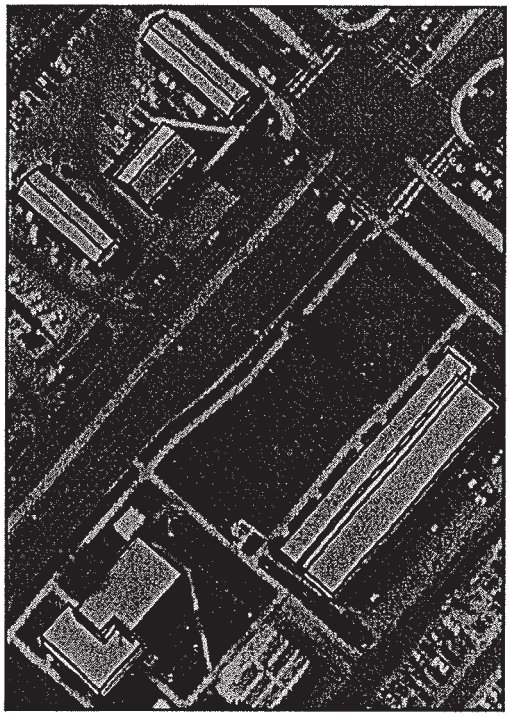
(c) Radt5s

(d) Radt5wob

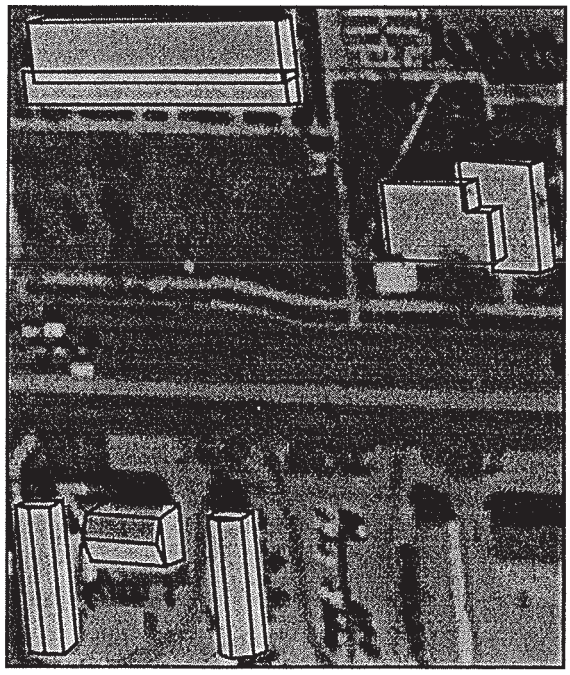
Figure 18: Scene 2 : Radt5



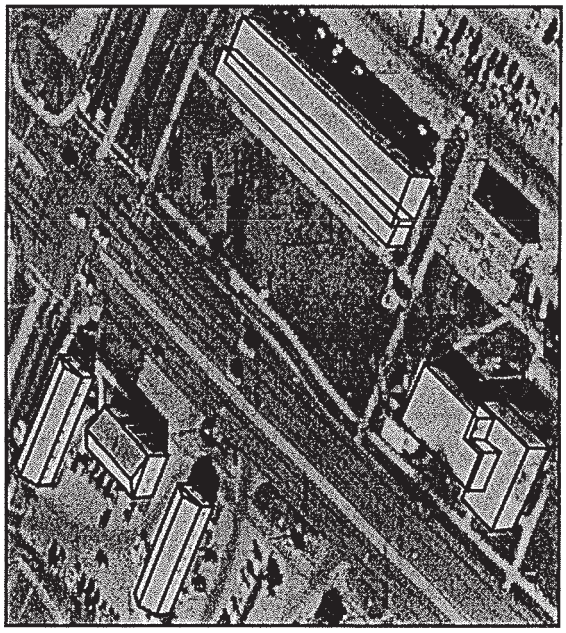
(a) Radt10



(b) Radt10s



(c) Radt10ob



(d) Radt10wob

Figure 19: Scene 3 : Radt10

Group A	Group B	Group C	Group D
radt9-A(m)	radt9-A(ap)	radt9-A(ap)	radt9-A(m)
radt9-A(ap)	radt9-A(m)	radt9-A(m)	radt9-A(ap)
radt10(m)	radt10(ap)	radt5(ap)	radt5(m)
radt10(ap)	radt10(m)	radt5(m)	radt5(ap)
radt5(ap)	radt5(m)	radt10(m)	radt10(ap)
radt5(m)	radt5(ap)	radt10(ap)	radt10(m)

Table 2: Measurements performed by four groups of subjects

be reduced. However, measuring a scene twice introduces another bias: a subject will become familiar with the scene, and should be able to perform the second set of measurement tasks with more efficiency. To account for this bias, half of the subjects measure a scene using the semi-automated process first, then the fully-manual approach. The other half perform the measurements in the opposite order. Finally, the order of the scenes are also swapped.

For the semi-automated measurements, a set of correspondence points were measured before the experiment in order to determine three of the four parameters used by SITECITY; *error propagation sensitivity* and *minimum* and *maximum elevation*. These parameters are computed using the methods described in Section 3.6. The same set of parameters are used throughout the experiments and they are shown in Table 3. One of the system parameters, *image sigma*, is set to 1.0 for all experiments. To ease the task of comparing building measurements, subjects were given a hard copy of the wire frame representations of the buildings in the scene and were told to measure buildings so that they had the same shape. The wire frames of the building in the hard copies were not overlaid over the images.

Image	Error Propagation Sensitivity	Minimum Elevation (meters)	Maximum Elevation (meters)	Number of Control Points
radt9	0.439580	−4.164819	14.212819	6
radt9ob	0.590738	−4.164819	14.212819	6
radt9s	0.306422	−4.164819	14.212819	6
radt9wob	0.488199	−4.164819	14.212819	6
radt5	0.622361	−5.546944	17.085346	8
radt5ob	0.596222	−5.546944	17.085346	8
radt5s	0.329514	−5.546944	17.085346	8
radt5wob	0.651511	−5.546944	17.085346	8
radt10	0.726335	−10.387142	16.697507	7
radt10ob	0.845945	−10.387142	16.697507	7
radt10s	0.657902	−10.387142	16.697507	7
radt10wob	0.708745	−10.387142	16.697507	7

Table 3: Parameters used in the semi-automated measurements

4.3 Establishing the Ground Truth

In order to evaluate the accuracy and precision of the delineation of an automated building detection system, a *ground truth* is needed to compare results. In previous studies [37; 50], the ground truth used in the automated building detection system was typically generated by one user. However, the *ground truth* generated by one user will vary from that generated by another user. Since a user is part of the the semi-automated system, it becomes difficult to assess the differences between semi-automated delineation and fully manual delineation

using this approach.

In our experiment, a building in an image is measured by numerous users. A building is measured by locating the image positions of building vertices in all available images and a measurement is the image position of a building vertex in an image. A *ground truth* in image space can be established by determining an average position of a point using multiple measurements. An advantage of generating a *ground truth* using multiple measurements is the ability to generate an error measure for each ground truth position. This error measure, expressed as a covariance matrix, allows us to quantify the accuracy and precision of any individual measurement compared to the mean, or the average position. To establish the average position and the covariance of a building point, corresponding building points from different measurements need to be determined. Since all buildings are measured with the same shape, it is trivial to establish an one-to-one correspondence between all the points. Once the correspondence of image point measurements is established, the average position and covariance of the point can be calculated. Figure 20 shows the covariance ellipse (drawn as a diamond shape) for measured building points on all four images for the radt9 scene. The average position of these measurements is the center of the ellipse (diamond) and the wireframe building shown in Figure 20 is drawn using the average positions. The size of the ellipse denotes the uncertainty of an image measurement and is drawn so that the boundary of the ellipse represents a distance of one standard deviation away from the mean. Figure 20 suggests that the measurement uncertainty of a point in an image is related to the *quality* of a corner in an image. When there is good contrast for a point, the covariance ellipse is small. This suggests that most users agree on the position of this point. When there is ambiguity, the size of the covariance ellipse increases, denoting more disagreement among users. If a point is occluded, the size of the covariance ellipse is larger than any other visible points.

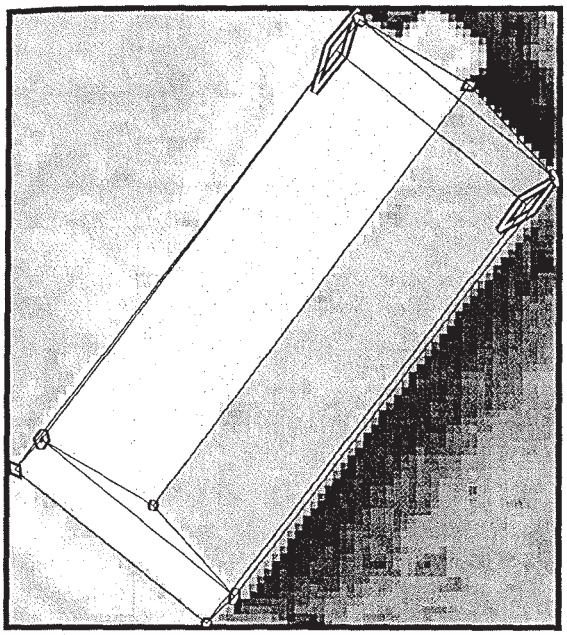
To determine the precision of a *ground truth* measurement, the area of the covariance ellipse of one standard deviation is defined to be the measurement uncertainty. If the area, or the measurement uncertainty, for a point is large, the group of measurements for that point is less precise. The occluded points are not used in any of the measurement statistics because it is meaningless to talk about precision of a measurement when the point is not measurable. The measurement uncertainties for all points are plotted as a cumulative histogram in Figure 23(a). Each point on the graph represents the percentage of points from the three test scenes having measurement uncertainty less than the values shown on the x axis. For example, the figure show that approximately 80% of measured points have measurement uncertainty of less than 10 square pixels.

4.4 Effects of Automated Processes on Measurement Accuracy

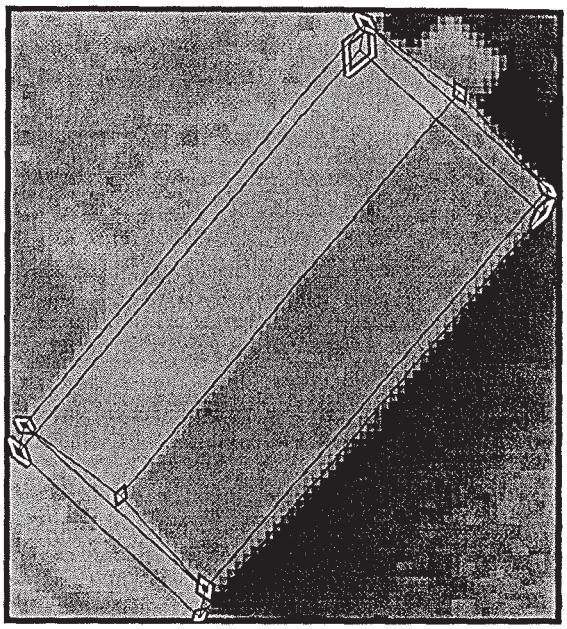
To evaluate the effect of the automated processes on the precision of measurements, the measured points are partitioned according to the method of measurements, fully manual or semi-automated. The measurement uncertainty for these two sets of measurements are shown in Table 4 and the average measurement uncertainty for each scene and method is plotted in Figure 23(b). The overall average measurement uncertainty, 10.48 square pixels, for all visible points is quite small, which corresponds to the area of a circle whose radius is 1.8 pixels. The radius of the equivalent circle of the covariance ellipse is another way to visualize the measurement uncertainty . Comparing to other scenes, measurement uncertainty for points in the radt10 scene is extremely large. This can be explained by noting that many corner points on the roofs are not visible due to poor imaging and photometry, especially in images radt10ob and radt10wob.

Our results show that the semi-automated system tends to produce more consistent measurements than the fully manual approach, because the average and standard deviation of measurement uncertainty for points measured using the semi-automated system is consistently less than the fully-manual system. This means that there are less variations in measurements performed by different users when the semi-automated system is used and does not necessary mean that the measurements are better.

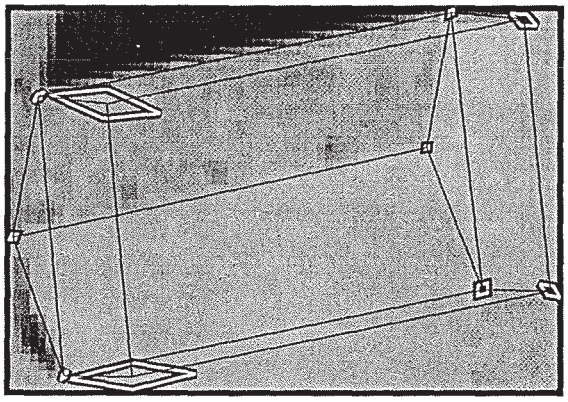
However, this consistency might be attributed to a bias introduced by the semi-automated system. To evaluate possible bias of the measurements between the semi-automated and the fully manual approaches, the distance vector between the average positions estimated by the semi-automated and fully manual system is used. For each point in the scene, the distance vector between the position of the same point measured by a subject using the semi-automated and the fully manual approaches are computed and plotted as a scattergram for each subject in Figures 21 and 22. The differences between the average positions of the semi-automated and fully manual measurements for all subjects are plotted in Figure 23(c).



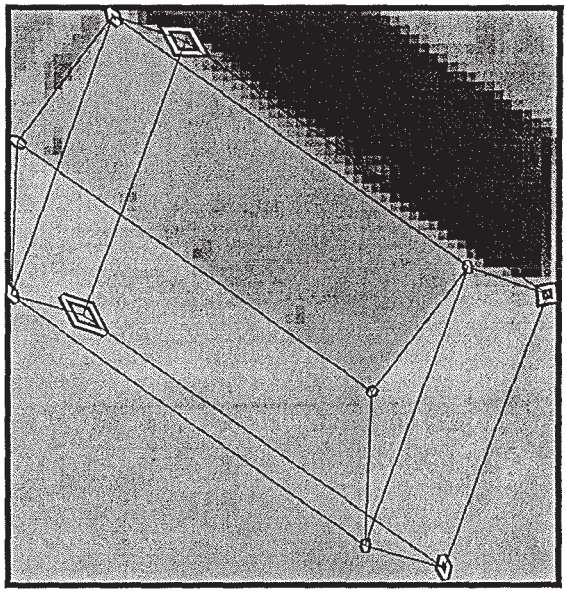
(a) Radt9-A



(b) Radt9s-A

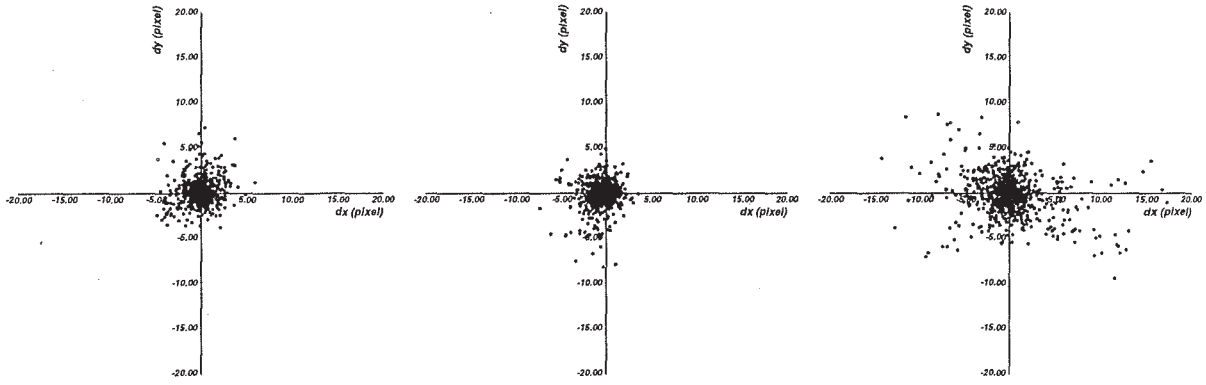


(c) Radt9ob-A

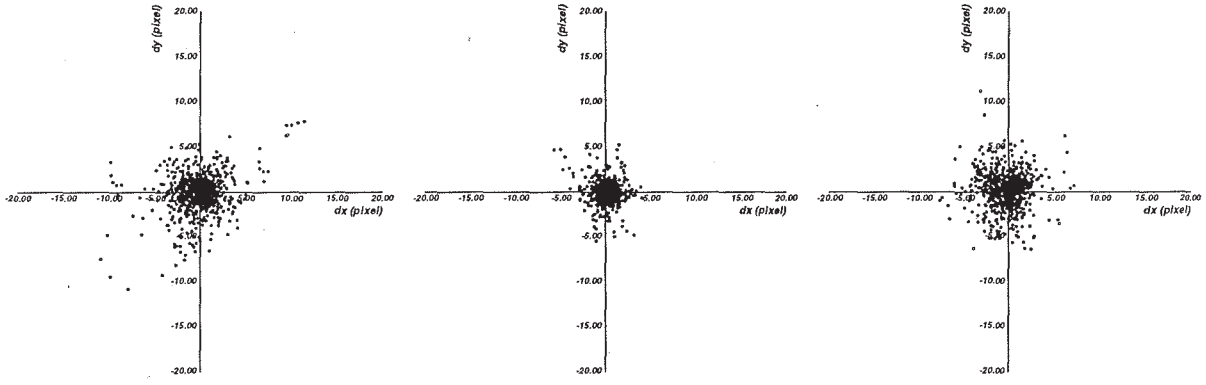


(d) Radt9wob-A

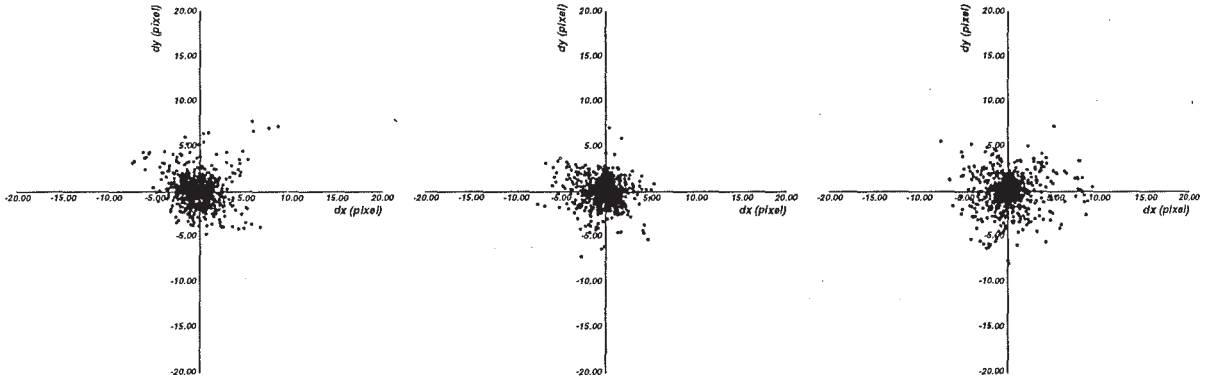
Figure 20: Covariance Ellipse For Radt9-A



(a) A1 (b) A2 (c) A3

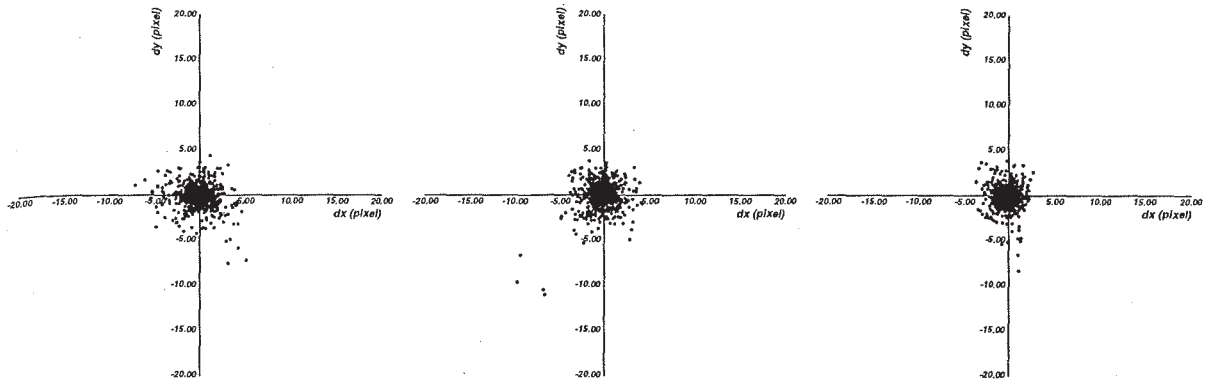


(d) B1 (e) B2 (f) B3



(g) C1 (h) C2 (i) C3

Figure 21: Difference vectors between fully manual and semi-automated measurements for each subject



(a) D1 (b) D2 (c) D3

Figure 22: Difference vectors between fully manual and semi-automated measurements for each subject

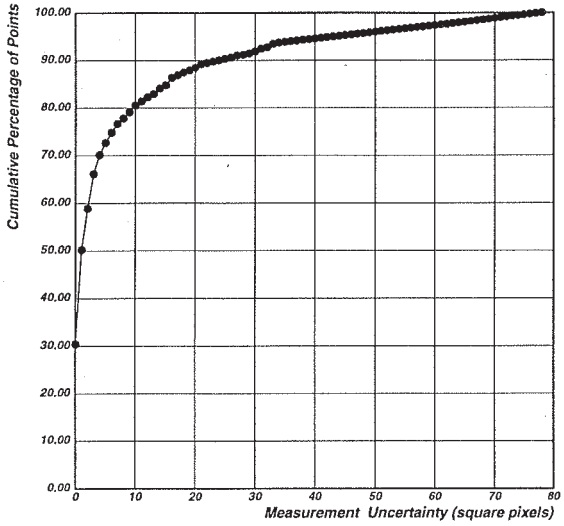
The distribution of the distance vector appears to be evenly distributed and no significant bias can be observed. Furthermore, the absolute distances between these two sets of measurements are plotted as a histogram in Figure 23(d); each rectangle represents the percentage of points whose distance between the semi-automated and the fully manual measurements is less than the distance shown in the x axis. This figure shows that approximately 80% of all measured points differs by less than 1 pixel. The few points with large differences consists of images points that are occluded by shadows, trees or other buildings, or the locations of the image points that are hard to measure due to poor image quality. These variations are to be expected. Based on these results, we conclude that there is no significant differences in the measurement accuracy between the semi-automated and fully manual system.

Scene	Sample Size	Average	Standard Deviation	Median
semi-automated	701	7.06(1.50)	19.16(2.47)	1.30(0.64)
fully-manual	701	11.44(1.91)	39.20(3.53)	2.10(0.82)
Total	703	10.48(1.83)	29.04(3.04)	1.97(0.79)
radt10(ap)	244	12.58(2.00)	24.60(2.80)	2.81(0.95)
radt10(m)	244	22.45(2.67)	63.34(4.49)	3.13(1.00)
radt10 scene	244	20.61(2.56)	46.02(3.83)	3.56(1.06)
radt5(ap)	421	4.32(1.17)	15.27(2.20)	1.04(0.58)
radt5(m)	421	5.56(1.33)	10.46(1.82)	1.93(0.61)
radt5 scene	423	5.25(1.29)	9.36(2.98)	1.82(0.76)
radt9(ap)	36	1.64(0.72)	2.50(0.89)	0.72(0.48)
radt9(m)	36	5.65(1.34)	17.68(2.37)	1.15(0.61)
radt9 scene	36	3.33(1.03)	7.39(1.53)	0.92(0.54)

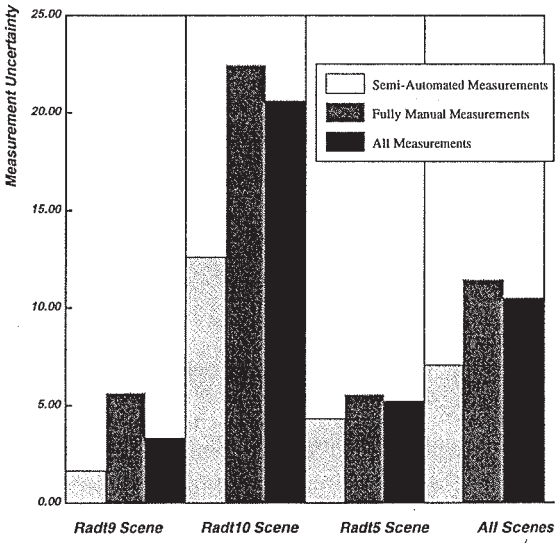
Table 4: Measurement uncertainty in square pixels (radius of the equivalent circle in pixels)

4.5 Performance Analysis of Automated Processes

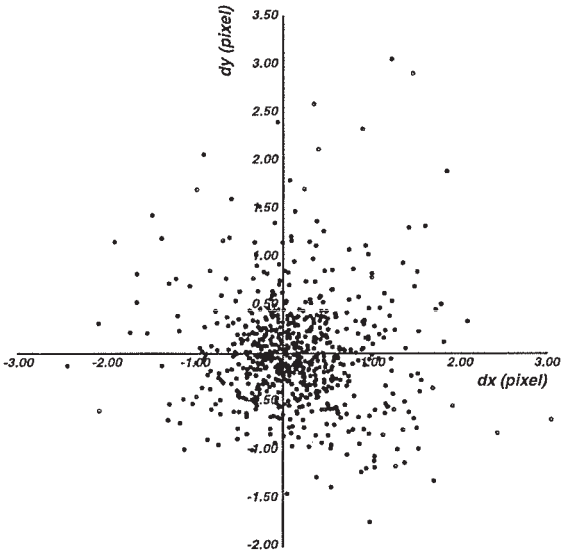
In this section, the performance of the automated processes is considered. Since the verification component described in section 3.1.1 is used to select and validate building object hypotheses for all automated processes, a detailed analysis concerning the sensitivity and performance of the verification component is presented first.



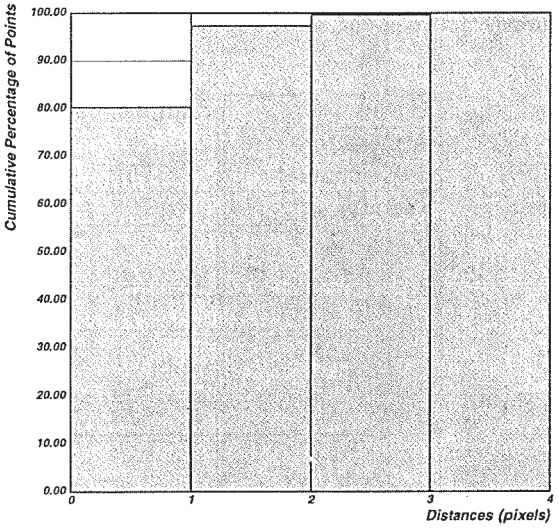
(a) Measurement uncertainty for all points



(b) Average measurement uncertainty for each scene and method



(c) Difference vectors between fully manual and semi-automated measurements



(d) Distances between fully manual and semi-automated measurements

Figure 23: Analysis of measurements precisions and bias

Next, the performance of the floor and peak edge detection processes is evaluated by using the manually measured roofs as the starting point. Finally, the epipolar matching process is evaluated using the manually measured buildings. The inputs to the automated processes are extracted from measurements performed by all subjects. The automated copy process is not evaluated for two reasons. First, the epipolar matching and automated copy share the same components, and hence the evaluation of the epipolar matching process and the verification component should provide a reasonable assessment of the automated copy process. Secondly, the automated copy process requires a cue measured by a user. The size and shape of the cues can influence the results of the automated copy process. This variation in the shape and size of the cues is difficult to control in the process of evaluation.

4.5.1 Analysis of the Verification Component

The goal of the verification component is to compute a confidence value for a building hypothesis, and the evaluation of the verification component is accomplished by comparing confidence values for the *ground truth* and *false hypotheses*. The evaluation is broken down into three categories: *the ability to discriminate*, *sensitivity to measurement error* and *sensitivity to displacement error*. *The ability to discriminate* category evaluates the ability of the verification component to differentiate between *ground truth* building objects and false hypotheses. This is determined by comparing the confidence values of the *ground truth* building objects and randomly generated *false* building objects. The next two categories evaluate the sensitivity of the verification component due to differences in the delineation of the building object. The *sensitivity to measurement error* category analyzes the response of the verification component to differences in the position of individual building vertices. The *sensitivity to displacement error* category determines the effect of a rigid body translation on the verification component. Detailed analyses of these three categories are presented below:

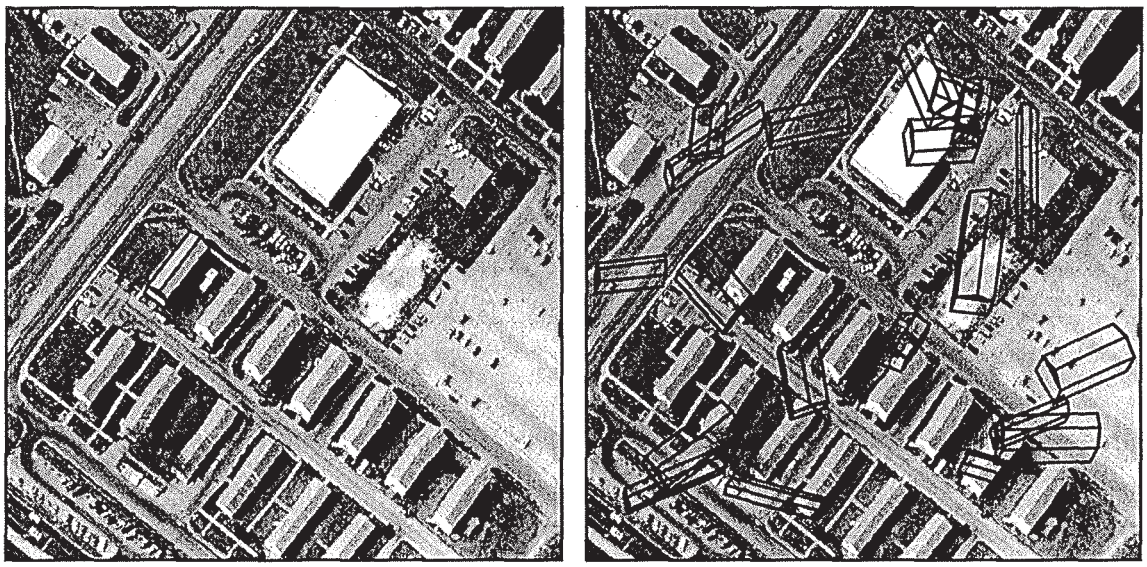
1. **The Ability to Discriminate**

   A perfect verification process should be able to select the correct building hypothesis among a set of false solutions. In other words, the verification function that computes the confidence of a hypothesis should give the maximum response (or maximum confidence) if the hypothesis is correct, and the minimal response if the hypothesis is false. The maximum confidence for the verification component in SITECITY is 4.0 and the minimum confidence value is 0.0 as described in Section 3.1.1.

   For this experiment, a set of manually measured building objects for scene 2 (Figure 18) and scene 3 (Figure 19) are used. For each manually measured building, 20 buildings of the same building type are randomly generated and placed in the image. Figure 24(b) shows a set of 20 random hypotheses generated using the manually measured building in Figure 24(a). The size of the randomly generated buildings can range from one half to twice the size of the manually measured building, and the position and orientation of the generated buildings are determined randomly. The random hypotheses do not overlap any of the manually measured buildings by more than 50% in image space. The confidence values computed using the verification component for the random hypotheses and the corresponding manual measurements for all images of the radt5 and radt10 scenes are plotted in Figures 25 and 26. For each trial, the confidence values of a manually measured building and a random hypothesis are computed and plotted.

   For the ideal verification component, the separation between the confidence values computed using the manual measurements and the random hypotheses should be maximum. Even though none of the graphs exhibits the ideal behavior, the separation between the two sets of confidence values is clearly distinct. The result of this experiment is summarized in Table 5, where the average and standard deviation of the confidence response for both the manual measurements and the random hypotheses are shown.

   Figures 25 and 26 show the comparison of the computed confidence values for the manual measurements and the random hypotheses for both the radt5 and radt10 scenes. These two figures show the dependency of the verification component on images and scenes. While confidence values for the random hypotheses concentrates around 1.0 for both scenes, the confidence values for the manual measurements depends on both the images and the objects. For the radt5 scene, the confidence values for the



(a) Manually measured building in radt5 image

(b) 20 random hypotheses

Figure 24: An example of random building hypotheses for the discrimination experiment

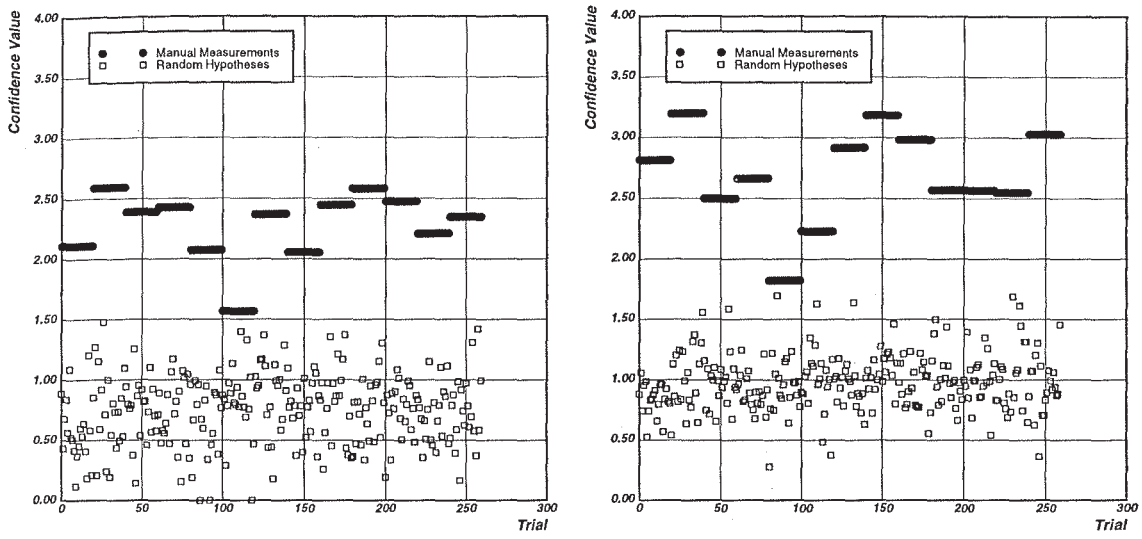
manual measurements varies between 1.5 and 3.0. It performed uniformly worse on the radt10 scene. The verification component will not be able to differentiate the manually measured buildings whose confidence values is below 1.5 from the random hypotheses shown in Figure 26. These poor confidence values can be explained by noting that inter-object occlusions, where one building object occludes all or parts of another building object, exist for those buildings in the radt10 scene. Since the verification component does not account for inter-object occlusions, it is expected to have worse performance with this scene. In addition, the image quality of the radt10 images is worse than the radt5 images. The buildings are blurry and the contrast is poor. The poor performance of the verification component on the radt10 scene is reflected later in the evaluation of the automated processes and the usability analysis.

The results of this experiment allow us to identify a threshold for determination of the good versus bad building hypotheses and problems with the verification component. While the verification component produced consistent confidence values for the random hypotheses, the confidence values for the manually measured buildings depends on a variety of factors. Collecting and analyzing a set of manually measured buildings with poor confidence values enables us to understand and improve the verification component. A threshold of the confidence value that discriminates between the good and bad building hypotheses involves tradeoffs. Figures 25 and 26 suggested that a threshold value between 1.5 and 2.0 will retain most of the good hypotheses while removing majority of the bad hypotheses. For SITECITY, we have decided to be cautious in not allowing too many false hypotheses and set the threshold value to be 2.0.

2. **Sensitivity to measurement error**

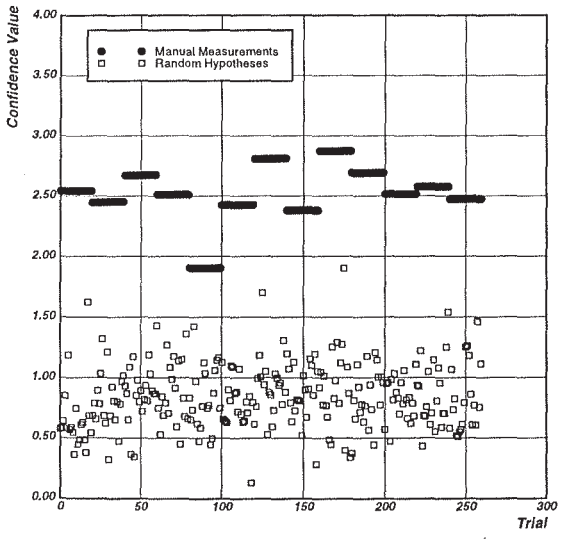
There are often variations in the placement of the measured points for both the manual and automated processes. There variations can be caused by the resolution of the image, the differences in the edge operators and the inherent differences between human operators. An ideal verification component should be able to tolerate small perturbations in measurements.

To evaluate the sensitivity of the verification component due to measurement variations, every point on the manually generated building object is perturbed randomly. The perturbed object is evaluated

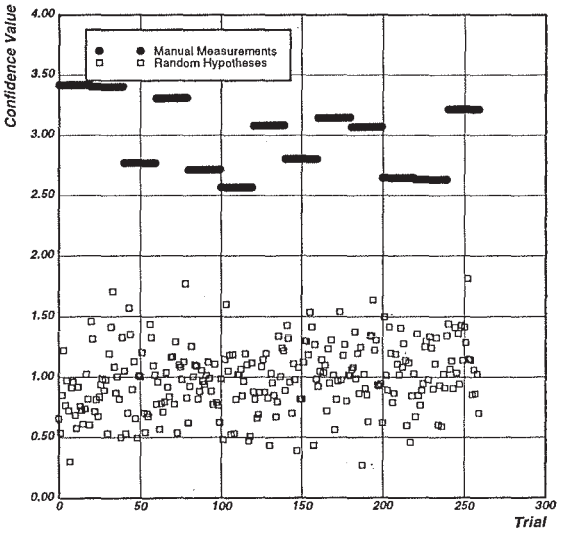


(a) Radt5

(b) Radt5ob

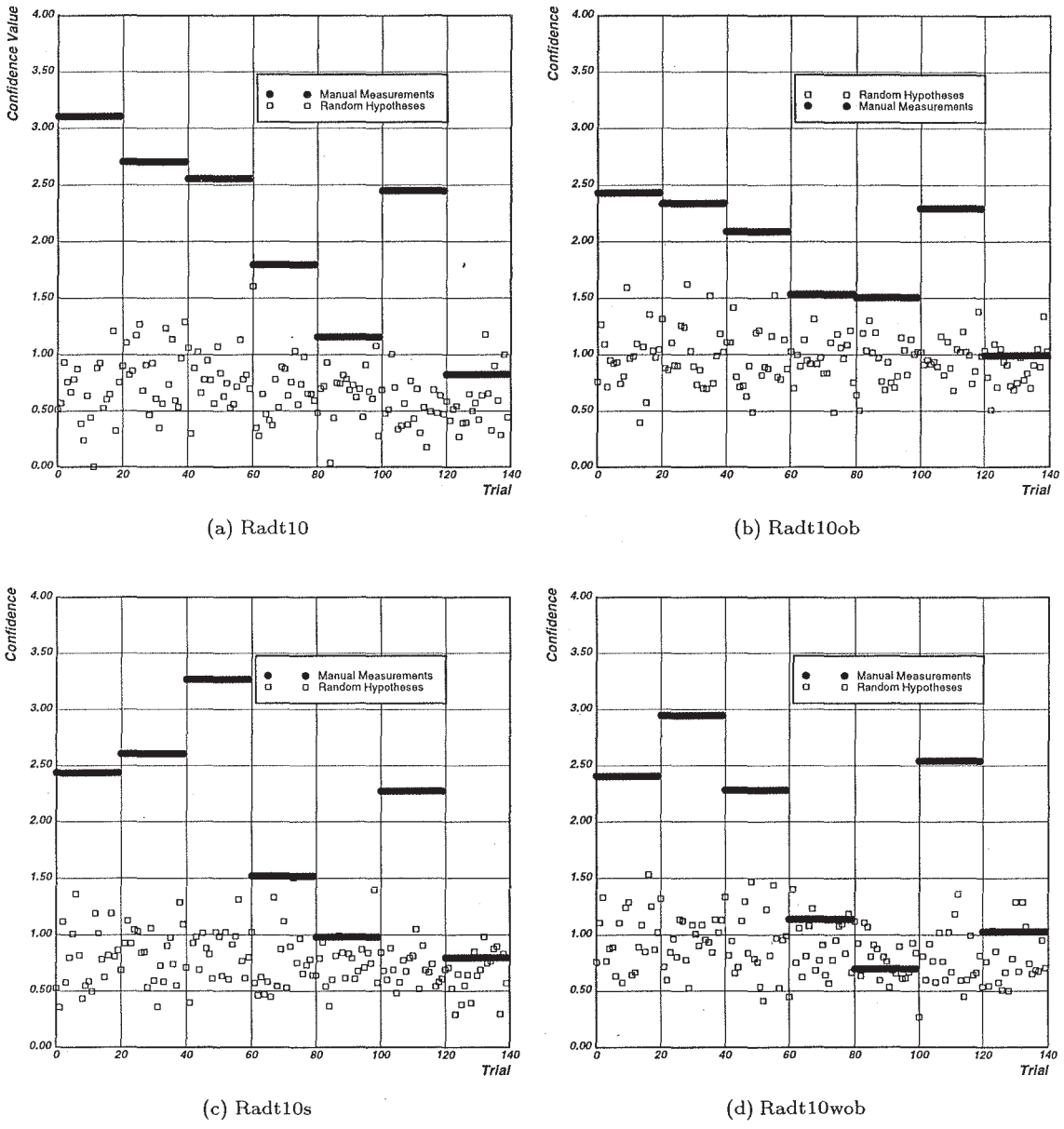


(c) Radt5s



(d) Radt5wob

Figure 25: Confidence value computed using the verification component on ground truth buildings and random hypotheses for the radt5 scene



(a) Radt10

(b) Radt10ob

(c) Radt10s

(d) Radt10wob

Figure 26: Confidence value computed using the verification component on ground truth buildings and random hypotheses for the radt10 scene

Image	Ground Truth Average Confidence (Standard Deviation)	Random Hypotheses Average Confidence (Standard Deviation)
radt10	1.90(0.82)	0.69(0.28)
radt10ob	1.52(0.51)	0.96(0.24)
radt10s	1.84(0.63)	0.78(0.28)
radt10wob	1.93(0.70)	0.90(0.28)
radt5	2.20(0.34)	0.74(0.30)
radt5ob	2.70(0.45)	1.00(0.25)
radt5s	2.41(0.30)	0.85(0.25)
radt5wob	2.95(0.23)	1.00(0.26)
Ideal Responses	4.00(0.00)	0.00(0.00)

Table 5: Summary of discrimination experiment

using the verification component to generate a confidence value. The confidence value of the perturbed object is plotted with respect to the perturbation distance to show the sensitivity of the verification component. An ideal verification component would return the maximum confidence value when the perturbation distance is 0, and monotonically decreasing as the distance increases.

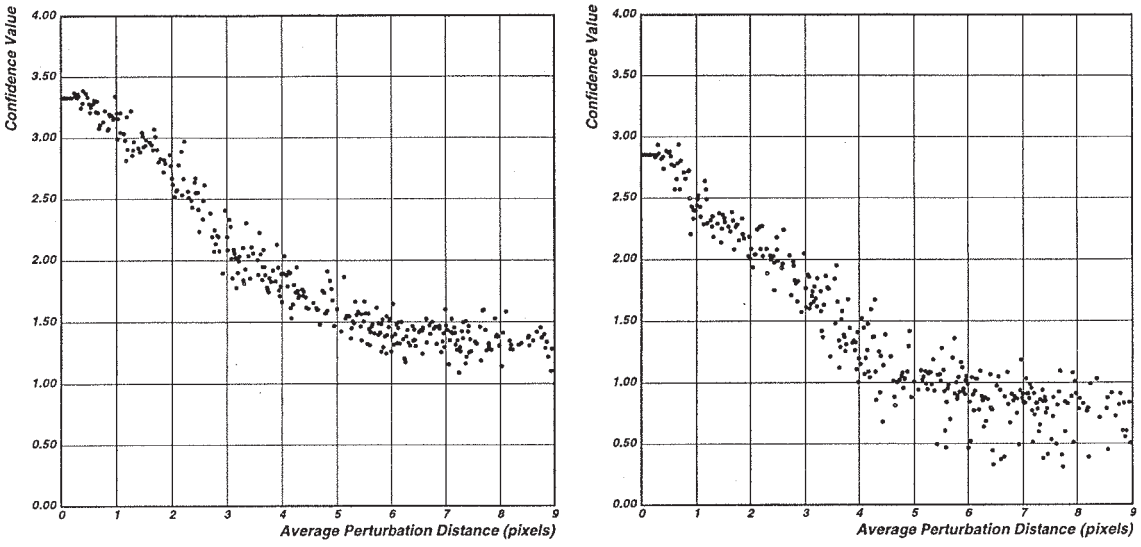
The building object shown in Figure 17 is used for this experiment. Every point on the building object is offset from the manually measured position with a random vector, for a total of 500 trials. The result of this experiment is shown in Figure 27. Each graph shows the average perturbation distance of the randomly perturbed building in the x axis and the confidence value of the perturbed building object in the y axis. These graphs show the response of the verification component in SITECITY to random perturbations of the *ground truth* object. Overall, the plots of distances versus the confidence values show the monotonically decreasing trend of the verification component despite local variations. It appears that as the distance grows larger, the confidence value approaches the average confidence values computed for the random hypotheses presented in Figure 25.

3. **Sensitivity to displacement error**

Another way to evaluate the sensitivity of the verification component is to consider the effect of a rigid translation of the building object in a image. What would be the behavior of the verification component if the measured object is translated some distance away from its measured position? In this experiment, the building object shown in Figure 17 is systematically translated in both the row and column direction in all four images. The range of the translation spans from -10 to 10 pixels away from the ground truth location. The confidence for the building object after each translation is computed with the verification component.

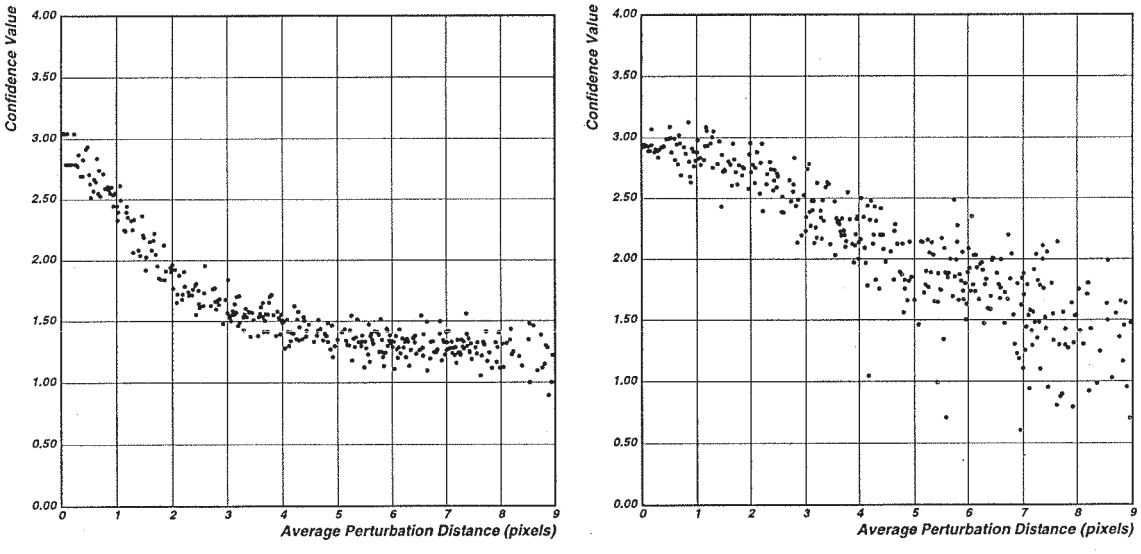
The responses of the verification component are shown in Figure 28 and the confidence values are encoded as a confidence image, where a brighter pixel value denotes higher confidence value. The confidence values are scaled linearly. The row and column coordinates of the image encode the translation distance in the row and column direction, with upper left corner having a translation vector $[-10, -10]$ pixels, and the lower right have values of $[10, 10]$. The intensity at center of the confidence image represents the confidence value of the original manually measured buildings without modifications. Ideally, the confidence values should monotonically decrease away from the center of the confidence image for an isolated building.

Figure 28 shows the result of this experiment on four images for the peak roof building object shown in Figure 17. The original peak roof building is drawn over the confidence images. For radt9-A, radt9ob-A and radt9s-A, the maximum confidence responses occur at $[1, 0]$, $[0, 0]$ and $[0, -1]$ respectively. For radt9wob-A, the maximal confidence response occurs with a translation vector of $[2, 0]$. For all of the confidence images, the confidence values slope away from the center, the ground truth position. However, the steepness of the slope is direction dependent, and this artifact is visible in the confidence



(a) Radt9-A

(b) Radt9ob-A



(c) Radt9s-A

(d) Radt9wob-A

Figure 27: Confidence value computed using the verification component on a randomly perturbed ground truth building

images as an intensity *ridge*. This artifact can be easily explained by noting that the direction of the *ridge* corresponds to the longitudinal orientation of the peak roof building as it appears in the image. Building objects translated along this axis share more similar features than objects translated in another direction, and have higher confidence values.

These three analyses demonstrate the behavior of the verification component in SITECITY. The verification components are evaluated by comparing the computed confidence values for *ground truths* and *noise* using real images. These experiments show that the verification component can differentiate a *correct* building hypothesis from noise. However, since the verification component is only part of the automated processes used in SITECITY, the performance of these automated processes still needs to be evaluated.

4.5.2 Analysis of the Automated Process

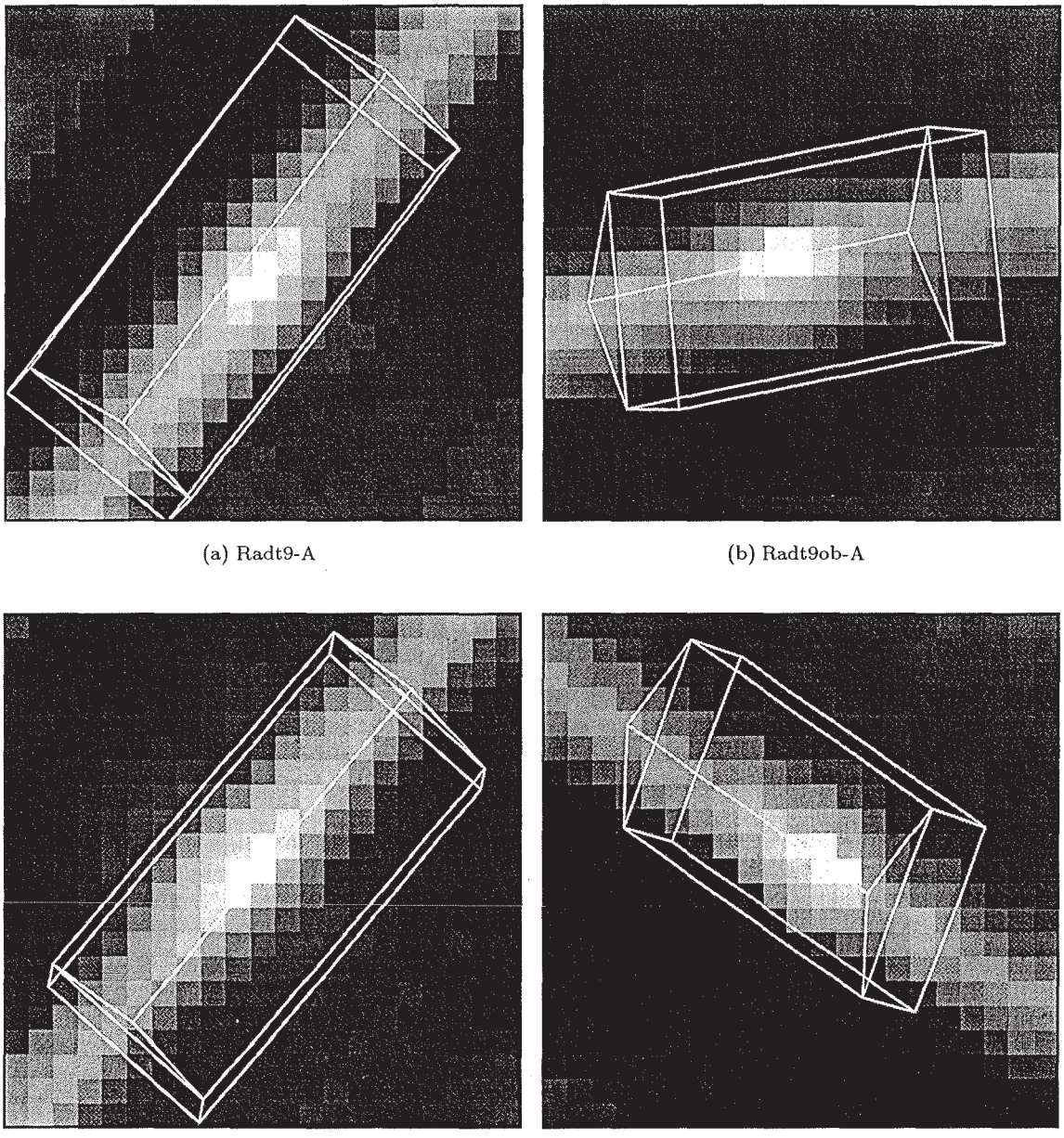
Unlike other performance evaluations for building detection and delineation systems, where the evaluation metric is typically on the building object level, the performance of automated processes in SITECITY is evaluated on individual building vertices. The topology of the user delineated building objects is verified to be correct. This requires the ability to match estimated point measurements with a *ground truth* point measurement.

For the analysis of the floor/peak detection process, the manually measured roofs are extracted from the manually measured buildings in all images. All manually measured buildings from all subjects are used. There are a total of 20 buildings in three scenes, and four images for each scene. Since there are 12 subjects, and each subject measures a scene twice, a total of 1920 roofs in 12 images were used. Out of the 1920 roofs, there are only 80 unique image buildings that were measured; this redundancy allows us to evaluate the sensitivity of the automated processes to different user inputs. For each manually measured roof in each image, the floor detection process is invoked using the extracted roof as the initial condition, to estimate the floor position. If the building is a peak roof building, the position of the peak edge is also estimated. Once a building is estimated in an image, the location of every visible image point estimated by the automated processes is compared with the *ground truth* measurements, the mean position and the covariance matrix (Section 4.3). Using the average and standard deviation values associated with each *ground truth* point, the accuracy of the point positions generated using the automated processes can be determined. Only the measurable points, visible in the image, estimated by the automated processes are used. For each image point estimated by the automated processes, the distance to the expected *ground truth* position, in terms of both the number of standard deviations away from the mean, and the Euclidean distance in terms of pixels, is computed.

The Euclidean distance measures the *absolute* accuracy of a point measurement in relationship to the *ground truth* position. On the other hand, the number of standard deviations measures the *relative* accuracy of a point measurement that accounts for the expected human performance. If a group of subjects can measure a point with good precision (*i.e.*, there is little doubt about the location of the point) then the automated processes are expected to return the same measurement. However, if a group of subjects measure a point with a large covariance (*i.e.*, they do not agree on the location of the point) then we do not expect the automated processes to be more accurate than an average user.

Figure 29 shows some examples contrasting the two distance measures. The mean positions and the ellipses representing 1 standard deviation for both points A and B are shown. Measurements 1A and 2A estimate the position of point A and have the same distance from the mean position of point A. The accuracy in terms of the Euclidean distance will be the same for both measurements. However, in terms of the number of standard deviations, measurement 1A is about 4σ away from the mean and measurement 1B is only about 0.5σ away from the mean. The Euclidean distance measure suggests that both measurements are equally good, but compared to the performance of user measurements, measurement 2A is significantly better than 1A.

For evaluation of a semi-automated system where users are involved, the measurements that agree with users perception should be considered better. For point A, users disagree on the location of the point along the major axis of the standard deviation ellipse, but agree about the location of the point along the minor axis. Since measurement 2A agreed with users on this respect, we expect the likelihood of a user to correct measurement 2A to be less than measurement 1A. For point B, the Euclidean distance of measurement 1B



(a) Radt9-A

(b) Radt9ob-A

(c) Radt9s-A

(d) Radt9wob-A

Figure 28: Confidence value computed using verification component on translated ground truth building. Brighter value corresponds to higher confidence.

to the mean position of point B is twice the distance of measurements 1A and 2A to the mean position of point A. Yet, the standard deviation measure of measurement 1B would be better than measurement 1A. This is because the position of point B is ambiguous in the image.

Using the number of standard deviations to compare the accuracy treats the automated processes with the same standard as human subjects; the automated processes need to perform as well as humans. At one extreme, for a point where there is no disagreement on its position, then any deviation in measurements using the automated processes will result in an infinite number of standard deviations away from the expected position, and we consider that the automated processes failed to measure the position correctly, even if the Euclidean distance of the measurement is less than 1 pixel away from the expected position. On the other hand, if users cannot agree on the position of a point due to poor images, then measurements made by the automated processes that is within one standard deviation of the expected position should be considered good regardless of Euclidean distance because there is no consensus in which to compare the measurements.

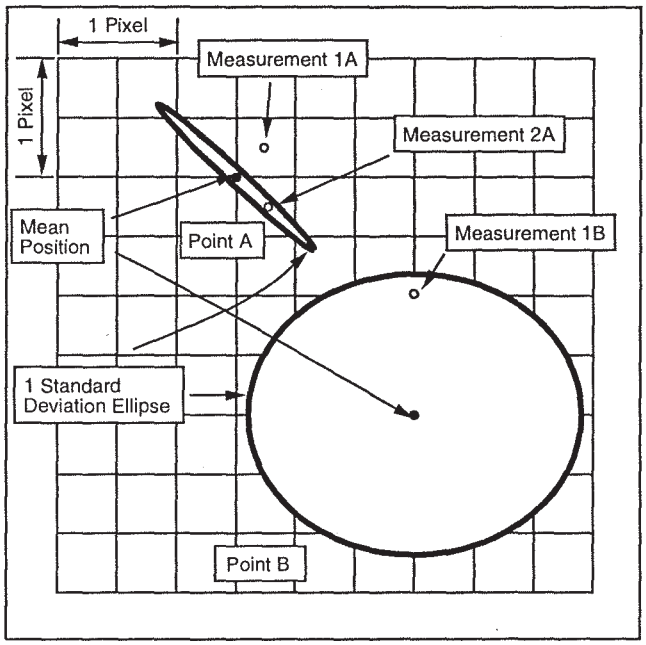
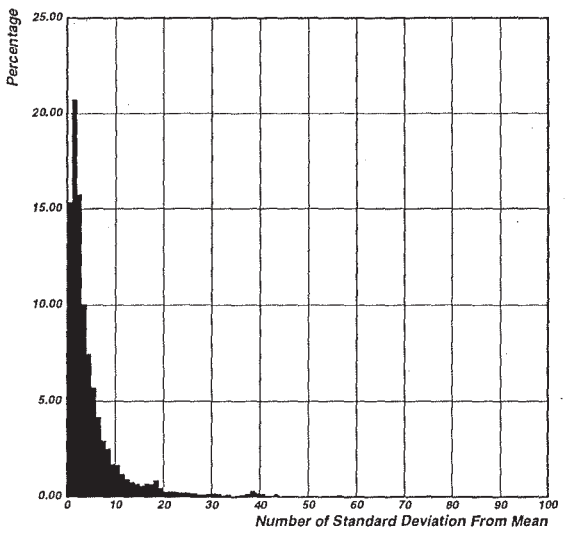


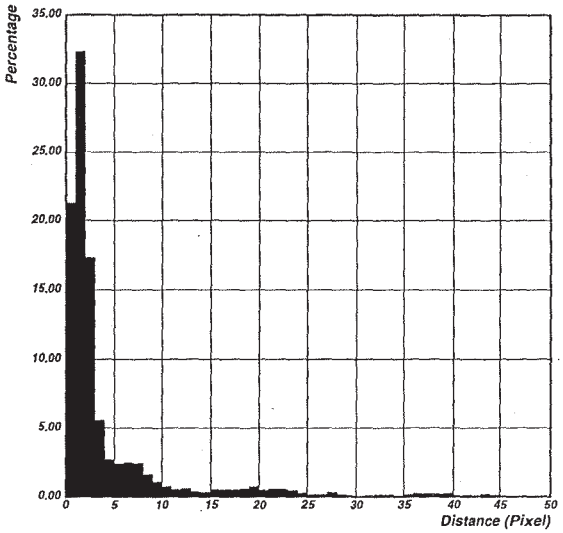
Figure 29: Example contrasting Euclidean distance and number of standard deviation for assessing accuracy of measurements

Both Euclidean distance and standard deviation distance are used to evaluate the performance of the automated processes. This evaluation process considers only visible vertices on building objects in images. In Figure 30(a), each bar denotes the percentage of points whose position estimated by the automated processes is between $n-1$ and n standard deviations away from the mean position. Figure 30(b) shows the percentage of point positions estimated by the automated processes that are between $n-1$ and n pixels away from the mean. The results from these two figures can be summarized in Tables 6 and 7. These tables show the result of the floor/peak detection process for each scene and image. The image column shows the image where the floor/peak detection was performed, while *npts* is the number of measurable points estimated by the automated processes. The remaining columns shows the cumulative percentage of points within N standard deviation of the manual measurements. The cumulated percentage in terms of point errors are plotted in Figures 30(c) and 30(d) for each image.

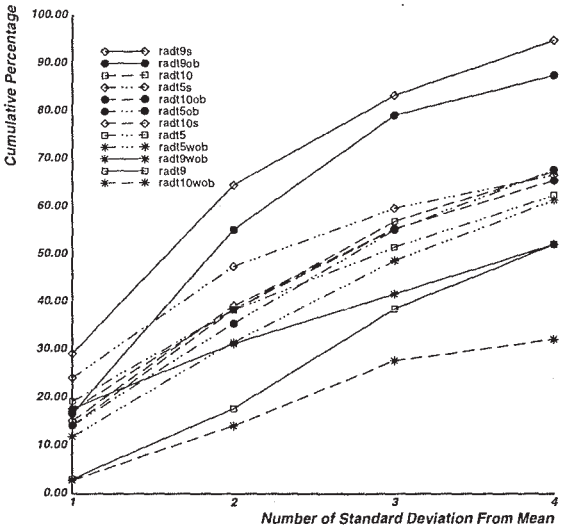
In Section 4.5, we analyzed the sensitivities of the verification component to various inputs. The same type of sensitivity study, where we consider performance variations due to different initial conditions, can be performed on the automated processes. Ideally, the automated processes should return the same result for an average user.



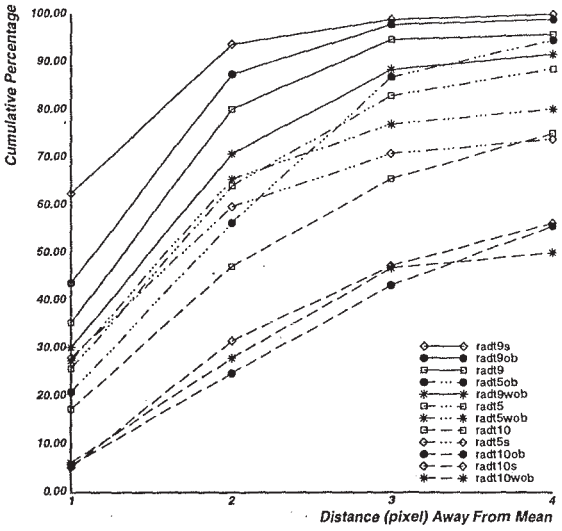
(a) Percentage of point errors (in terms of σ)



(b) Percentage of point errors (in terms of distance)



(c) Cumulative percentage of point errors (in terms of σ)



(d) Cumulative percentage of point errors (in terms of distance)

Figure 30: Performance of floor/peak processes

Image	npts	$\% \leq 1\sigma$	$\% \leq 2\sigma$	$\% \leq 3\sigma$	$\% \leq 4\sigma$
radt9	96	3.13	17.71	38.54	52.08
radt9ob	96	16.67	55.21	79.17	87.50
radt9s	96	29.17	64.58	83.33	94.79
radt9wob	96	17.71	31.25	41.67	52.08
radt9 scene	384	16.67	42.19	60.68	71.61
radt5	864	19.21	38.31	51.50	62.38
radt5ob	1008	14.19	35.52	55.16	67.66
radt5s	1128	24.11	47.52	59.66	66.58
radt5wob	1296	11.88	31.56	48.69	61.34
radt5 scene	4296	17.11	38.04	53.65	64.41
radt10	576	14.24	38.37	56.94	67.36
radt10ob	504	17.46	38.49	55.36	65.48
radt10s	456	15.13	39.25	52.63	62.94
radt10wob	672	2.83	14.14	27.68	32.14
radt10 scene	2208	11.68	31.20	46.78	55.30
Total	6888	15.35	36.08	51.84	61.89

Table 6: Summary of floor/peak detection performances based on standard deviation

Image	npts	$\% \leq 1$ pixel	$\% \leq 2$ pixels	$\% \leq 3$ pixels	$\% \leq 4$ pixels
radt9	96	35.42	80.21	94.79	95.83
radt9ob	96	43.75	87.50	97.92	98.96
radt9s	96	62.50	93.75	98.96	100.00
radt9wob	96	30.21	70.83	88.54	91.67
radt9 scene	384	42.97	83.07	95.05	96.35
radt5	864	25.81	64.12	83.10	88.66
radt5ob	1008	20.93	56.35	87.00	94.64
radt5s	1128	28.10	59.75	70.92	73.94
radt5wob	1296	27.47	65.51	77.16	80.25
radt5 scene	4296	25.77	61.57	79.03	83.66
radt10	576	17.36	47.22	65.63	75.17
radt10ob	504	5.75	24.80	43.25	55.75
radt10s	456	5.26	31.58	47.37	56.36
radt10wob	672	6.25	27.98	46.88	50.15
radt10 scene	2208	8.83	33.02	51.04	59.24
Total	6888	21.30	53.61	70.95	76.54

Table 7: Summary of floor/peak detection performances based on distance

The input to the automated floor/peak detection process is the roof delineation. Since each building in the three experimental scenes has 24 different roof delineations (12 subjects measuring a same roof using two different methods), the results of the automated processes on the 24 roof delineations for the same building can be collected and analyzed. Each roof delineation returns a building hypothesis. Since the buildings are of the same shape, one-to-one correspondences can be established resulting in 24 measurements for each building point.

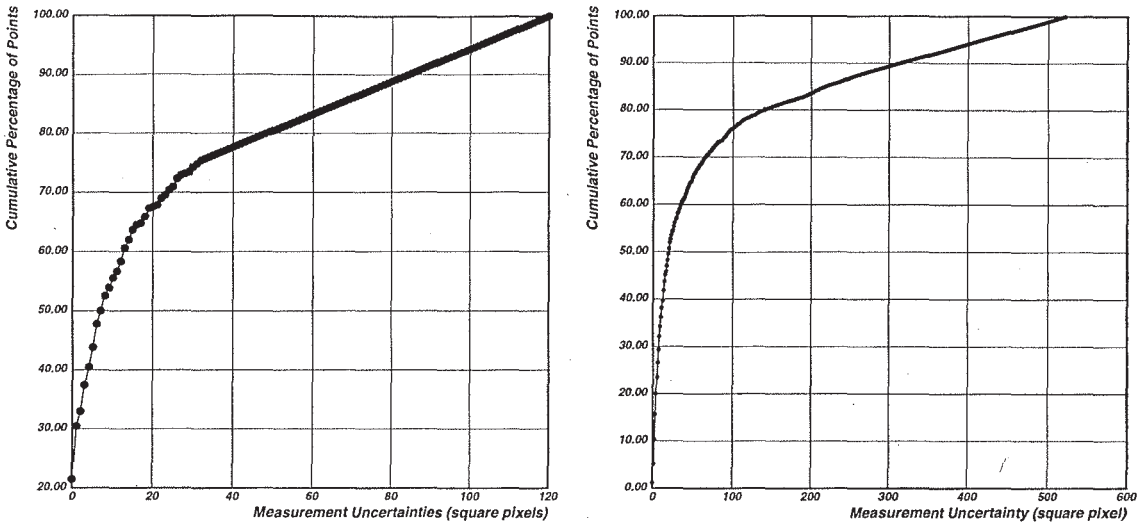
If the automated processes are tolerant of different user inputs, then the 24 measurements for each same building point should be at about the same location. The measurement uncertainty metric described in Section 4.3 can be used to assess the similarity of these measurements. If the automated processes are insensitive to the 24 different initial conditions in this experiment, then the measurement uncertainties for the building point should be small. The measurement uncertainty for other buildings in other scenes can be computed using the same method. In Figure 31(a) the cumulative percentage of points with measurement uncertainties of less than the value on the x–axis is plotted on the y–axis. The average measurement uncertainties for each scene and the overall results are shown in Table 8 and plotted in Figure 31(a). Overall, about 75% of estimated points have measurement uncertainties of less than 30 square pixels, which is equivalent to a circle whose radius is 3.1 pixels. These values suggest that the floor/peak detection process might be too sensitive to the initial conditions.

Scene	Sample Size	Average	Standard Deviation	Median
radt9	5	2.26(0.85)	1.30(0.64)	1.69(0.73)
radt9ob	4	0.64(0.45)	0.40(0.36)	0.71(0.48)
radt9s	5	0.88(0.53)	0.33(0.32)	0.77(0.49)
radt9wob	5	25.30(2.84)	25.17(2.83)	23.31(2.72)
radt9 scene	19	7.62(1.56)	16.11(2.26)	1.09(0.59)
radt10	29	76.17(4.92)	141.24(6.71)	12.62(2.00)
radt10ob	29	575.77(13.54)	937.68(17.28)	42.86(3.69)
radt10s	26	39.08(3.53)	41.97(3.66)	25.46(2.85)
radt10wob	29	252.20(8.96)	588.20(13.68)	9.76(1.76)
radt10 scene	113	241.03(8.76)	597.69(13.79)	19.40(2.48)
radt5	55	15.69(2.23)	20.66(2.56)	8.53(1.65)
radt5ob	55	11.84(1.94)	25.97(2.88)	1.77(0.75)
radt5s	58	86.06(5.23)	162.04(7.18)	13.96(2.11)
radt5wob	58	33.31(3.25)	91.81(5.41)	5.73(1.35)
radt5 scene	226	37.33(3.45)	99.70(5.63)	6.43(1.43)
All scenes	358	100.05(5.64)	357.20(10.67)	8.12(1.60)

Table 8: Measurement uncertainty for peak/floor detection processes in square pixels (radius of the equivalent circle in pixels)

The evaluation of the epipolar matching process uses the same metrics. To evaluate the epipolar matching process, manually measured buildings in one image are projected to other images and the epipolar matching process is performed to locate the object position in those images. The position estimated by the automated process is not manually corrected, to evaluate the performance of the automated process. For this experiment, all manually measured buildings from all subjects are used. For the three scenes used in the experiment, a total of 80 delineated buildings in image space (20 buildings in 4 images) are available. Since each user measures the building twice, there are a total of 160 independent building delineations in image space for each subject. Each building delineation in each image is projected to the other three images and matched, resulting in a total of 480 epipolar matching processes for each subject. Since there are 12 subjects, a total of 5760 epipolar matching processes are used for this experiment.

Once the building objects are matched, the analysis of the epipolar matching process is similar to the



(a) Floor/peak detection process

(b) Epipolar matching process

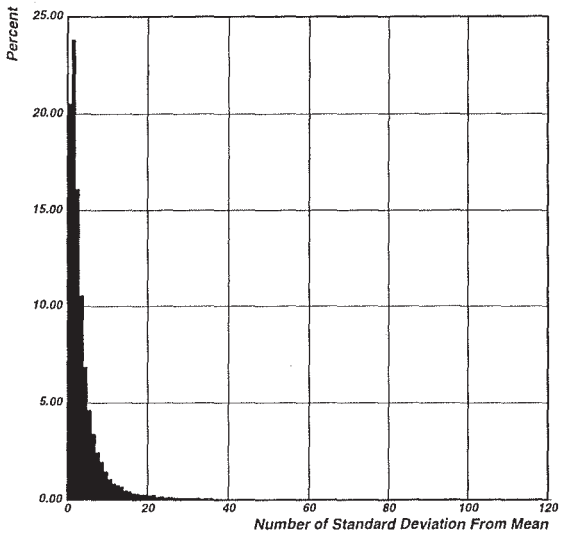
Figure 31: Histogram of measurement uncertainties for the automated processes

analysis for the peak/floor detection processes. The point estimation errors in terms of the number of standard deviations and distance for all points is plotted in Figures 32(a) and 32(b). The summary of the errors are shown in Tables 9 and 10, and plotted in Figure 32(c) and 32(d). The sensitivity of the epipolar matching process is shown in Table 11 and plotted in Figure 31(b).

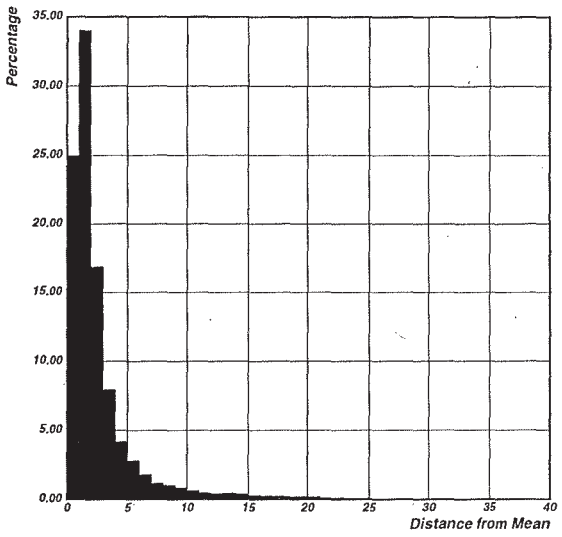
4.5.3 Other Experiments

The 12 images used for the evaluation of the automated processes in the previous section only represent a small fraction of all possible scenes and images. In order to have a good grasp of the capability of a system, it is important to test a system on a variety of scenes and images. However, the effort required to compile the manual measurement statistics presented in previous sections is impractical for a large number of data sets. Therefore, in this section, we present the performance of the automated processes on a variety of scenes in which the results from the automated processes are compared to manual measurements by one subject, and only the performance evaluation based on distance will be presented.

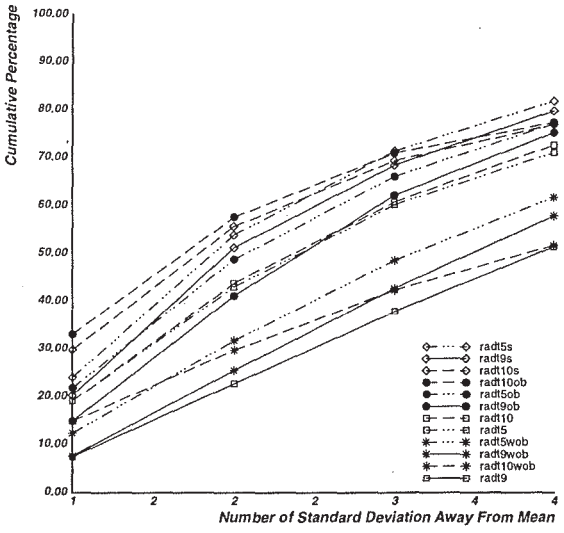
Figure 33 shows an image from each of the four additional scenes. Figure 33(a) shows the image j1 for the model board scene, distributed under the RADIUS program. For this scene, 8 images were used, j1 to j8. This is a complicated scene due to numerous occlusions and many structures on top of the buildings. For this scene, only buildings that are visible in all 8 images are measured. The model board scene is a small scale model of an industrial scene imaged using a light source at a close range to the models. The shadow verification components are not used because the shadows shown in these model board images violate our assumption that the light source (*e.g.* the sun) is an infinite distance away from the scene. The computed sun azimuth and elevation can vary as much as 20 degrees across some images. Figure 33(b) shows an industrial scene in the Avenches dataset provided by ETH, Zurich. The images for this scene are high resolution and contain buildings that do not correspond to the defined model type. Some of the peak roof buildings have overhanging roofs, and some buildings that appear to be flat roof buildings actually have sloped roofs. This scene presents a challenge to the system because it violates our assumptions about building shapes and allows us to determine the influence of inappropriate models on the automated processes. Four images (avenches5873, avenches5874, avenches5882 and avenches5883) are used in this scene. Figure 33(d) shows another image of the Fort Hood scene. Four images (fhov1627, fhov927, fhov525 and fhov727) are used for



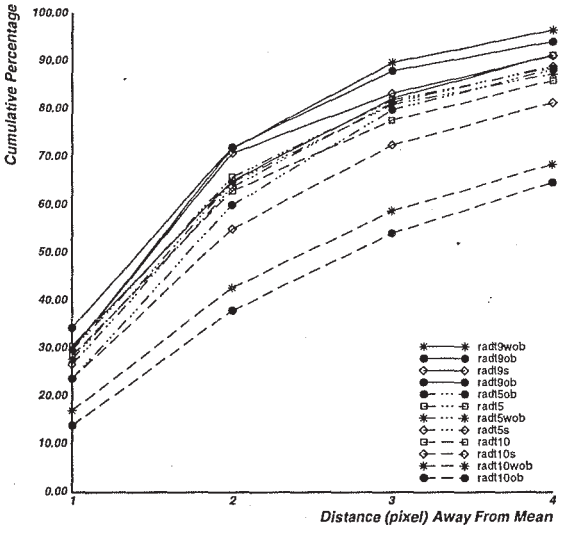
(a) Percentage of point errors (in terms of σ)



(b) Percentage of point errors (in terms of distance)



(c) Cumulative percentage of point errors (in terms of σ)



(d) Cumulative percentage of point errors (in terms of distance)

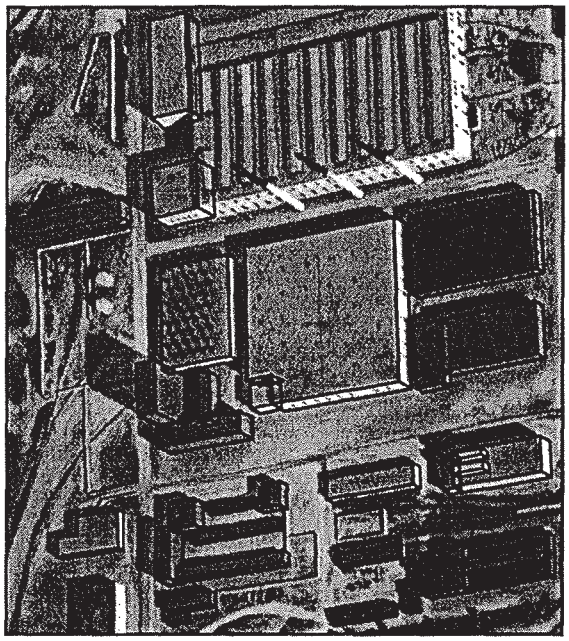
Figure 32: Performance of epipolar matching processes

Image	npts	$\% \leq 1\sigma$	$\% \leq 2\sigma$	$\% \leq 3\sigma$	$\% \leq 4\sigma$
radt9	576	7.47	22.74	37.85	51.39
radt9ob	576	14.93	41.15	62.15	75.17
radt9s	576	20.31	51.22	68.40	79.69
radt9wob	576	7.64	25.52	42.71	57.81
radt9 scene	2304	12.59	35.16	52.78	66.02
radt10	4032	19.12	43.77	60.74	72.59
radt10ob	3812	33.13	57.58	70.91	77.28
radt10s	3672	29.90	55.64	69.36	76.85
radt10wob	4236	14.90	29.77	42.35	51.68
radt10 scene	15752	23.89	46.11	60.27	69.10
radt5	5760	19.17	43.00	60.23	71.02
radt5ob	6479	21.86	48.76	66.06	77.03
radt5s	6768	24.10	53.87	71.35	81.74
radt5wob	7245	12.39	31.79	48.57	61.68
radt5 scene	26252	19.23	44.13	61.32	72.69
Total	44308	20.54	44.37	60.50	71.07

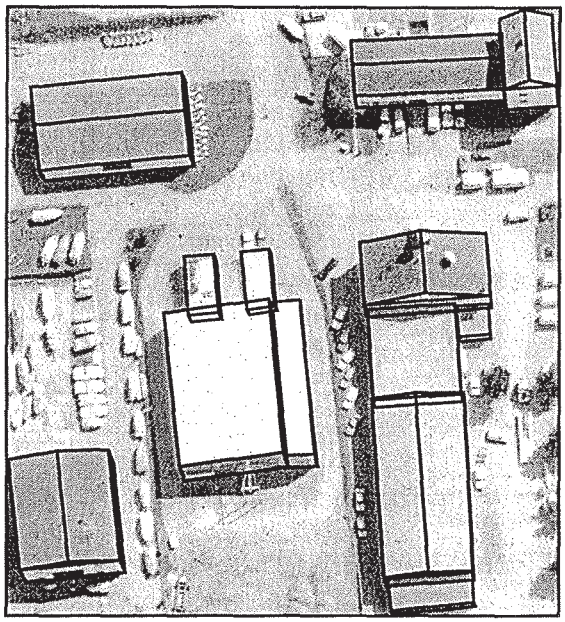
Table 9: Summary of epipolar matching performances based on standard deviation

Image	npts	$\% \leq 1$ pixel	$\% \leq 2$ pixels	$\% \leq 3$ pixels	$\% \leq 4$ pixels
radt9	576	30.56	65.10	82.29	91.32
radt9ob	576	34.38	72.05	88.02	94.10
radt9s	576	29.86	70.83	83.33	91.15
radt9wob	576	29.86	71.88	89.76	96.53
radt9 scene	2304	31.16	69.97	85.85	93.27
radt10	4032	28.52	63.00	77.75	85.94
radt10ob	3812	13.93	37.99	54.20	64.77
radt10s	3672	23.77	55.07	72.60	81.37
radt10wob	4236	17.04	42.78	58.85	68.53
radt10 scene	15752	20.80	49.66	65.77	75.07
radt5	5760	29.72	65.89	81.94	88.63
radt5ob	6479	23.74	60.07	79.92	88.27
radt5s	6768	26.71	63.84	81.44	88.95
radt5wob	7245	27.83	64.83	81.01	87.33
radt5 scene	26252	26.95	63.63	81.06	88.26
Total	44308	24.98	58.99	75.87	83.83

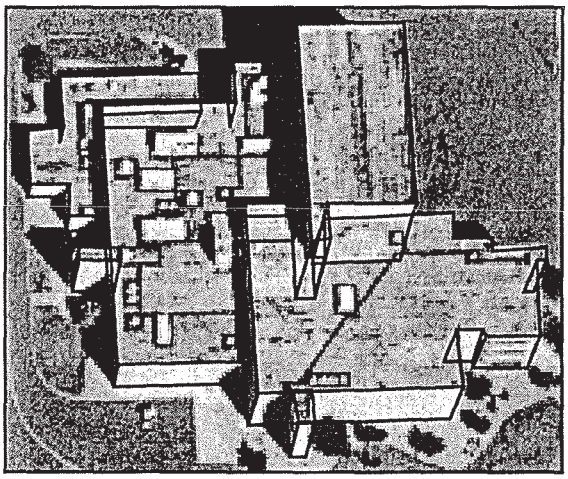
Table 10: Summary of epipolar matching performances based on distance



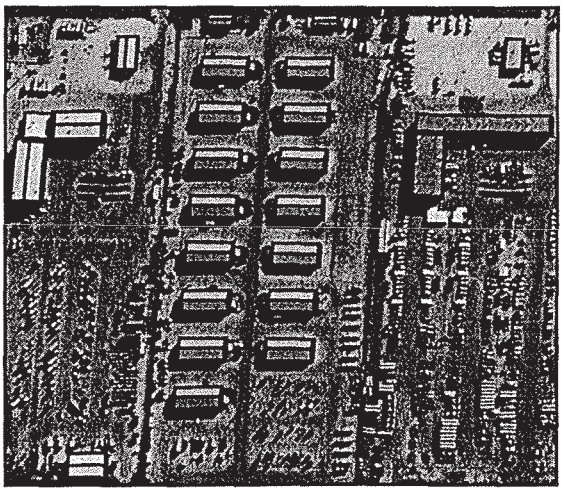
(a) Model board J



(b) Avenches industrial site



(c) Martin Marietta site (mmbld)



(d) Radt9

Figure 33: Other test scenes

Scene/Image	Sample Size	Average	Standard Deviation	Median
radt9	24	17.88	18.30	11.69
radt9ob	24	7.12	5.34	5.26
radt9s	27	58.21	78.54	8.51
radt9wob	27	7.43	6.25	5.62
radt9 scene	102	23.26	46.29	7.12
radt5	309	48.48	74.24	13.82
radt5ob	311	70.27	185.69	11.98
radt5s	318	38.91	78.32	9.26
radt5wob	318	52.83	73.06	21.22
radt5 scene	1256	52.55	113.56	13.83
radt10	183	98.78	181.27	14.05
radt10ob	183	1093.87	1739.64	245.14
radt10s	169	180.34	522.65	38.46
radt10wob	183	585.55	983.35	207.98
radt10 scene	718	495.67	1114.91	67.90
Total	2076	204.37	694.51	20.27

Table 11: Measurement uncertainty for epipolar matching processes (square pixels)

this scene. Figure 33(c) shows the image 74ov for a complex building in the Martin Marietta image set (mmbld), also distributed under the RADIUS program. The buildings are models as a composite of six rectilinear flat roof building objects. This scene has two images (74ov and 51ov). The parameters for these images are shown in Table 12.

The results of the roof/peak detection and the epipolar matching processes are shown in Tables 13 and 14, and Figure 34. Overall, the Avenches test set produced the worst result. This is most likely a function of the inappropriate building models used. Radt9 produced the best results for both the floor/peak detection and epipolar matching processes, most likely due to the good image quality and the simple building objects. The results for the floor detection process is surprisingly good for the Martin Marietta scene, considering that most of the building components are occluded. However, the shadows in the scene are clear and shadow verification might be a contributing factor to the performance of the automated processes.

4.5.4 Summary of Performance Evaluation of Automated Processes

Currently, the automated processes are affected by a combination of scene complexity, image contrast, view angles and users; the automated processes perform well when the scenes are simple with few occlusions and good contrast.

We expected that an oblique image will have better performance than a vertical image, because there can be more visible building facets in oblique images. However, our experiments do not show a clear advantage for oblique images; in fact, near vertical images seem to perform just as well, if not better, than some oblique images. One possible explanation is that while oblique images provide more information about the object, the search areas in the image also increase for the oblique image and increase the chance for mismatches. Another explanation for the performance on the near vertical images is the use of the shadow verification component. Building shadows can be a useful clue in vertical images [28], where walls and floors are not visible.

The evaluation of point estimation errors in terms of both the standard deviation and the pixel distance shows the need for automated processes that can align edges and corner points with subpixel accuracy. Overall, approximately 50% of the positions estimated by the automated processes are within 2 pixels of the manual measurements; more work is need to increase the performance of the automated processes. Users can often perform building delineation within subpixel accuracies by adjusting the display resolution. If the

Image	Error Propagation Sensitivity	Minimum Elevation (meters)	Maximum Elevation (meters)	Number of Control Points
j1	0.984998	−8.906343	57.839893	11
j2	0.954196	−8.906343	57.839893	11
j3	0.170762	−8.906343	57.839893	11
j4	0.496565	−8.906343	57.839893	11
j5	1.422947	−8.906343	57.839893	11
j6	0.110999	−8.906343	57.839893	11
j7	0.429516	−8.906343	57.839893	11
j8	1.318872	−8.906343	57.839893	11
fhov727	0.204139	−36.344445	−17.091910	4
fhov525	0.138896	−36.344445	−17.091910	4
fhov927	0.131330	−36.344445	−17.091910	4
fhov1627	0.199404	−36.344445	−17.091910	4
74ov	0.347015	−16.322330	48.709052	4
51ov	0.600973	−16.322330	48.709052	4
avenches5883	1.641601	−25.887208	17.155068	6
avenches5882	1.609350	−25.887208	17.155068	6
avenches5874	1.545972	−25.887208	17.155068	6
avenches5873	1.440986	−25.887208	17.155068	6

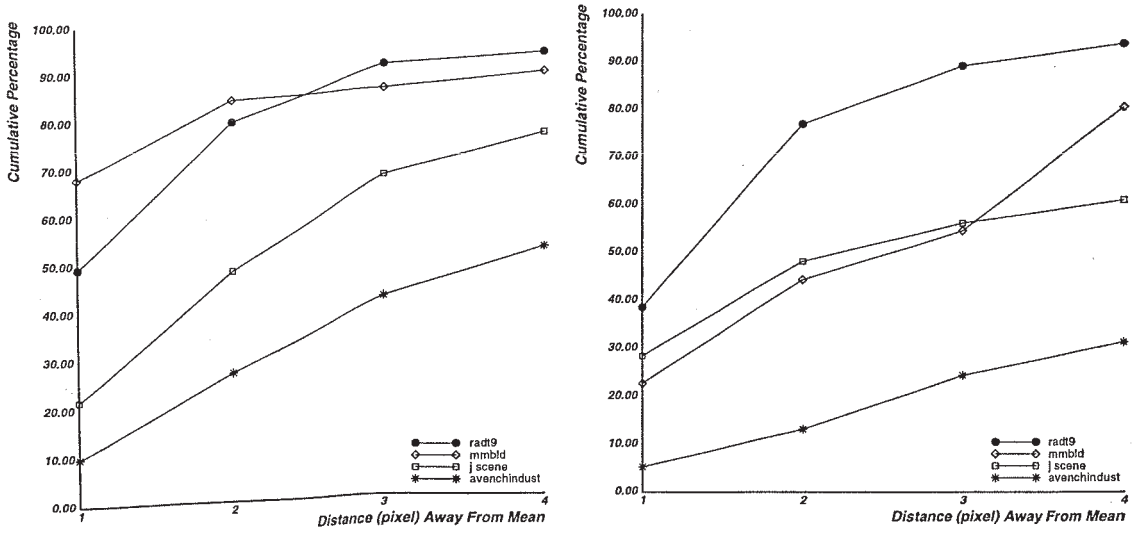
Table 12: Parameters used for test scenes

Image	npts	% ≤ 1 pixel	% ≤ 2 pixels	% ≤ 3 pixels	% ≤ 4 pixels
j1	236	25.85	56.78	66.10	73.31
j2	236	24.58	56.36	64.83	69.92
j3	215	6.05	27.91	53.02	72.56
j4	204	33.33	59.31	73.53	80.39
j5	206	6.31	31.07	64.56	70.87
j6	236	32.63	57.20	75.42	86.44
j7	212	12.26	37.26	64.62	74.53
j8	237	30.80	54.43	71.31	76.37
j scene	1782	21.83	47.98	66.78	75.59
radt9-fhov1627	226	57.08	94.69	98.67	99.12
radt9-fhov927	226	68.58	83.63	91.15	92.48
radt9-fhov525	212	4.25	41.98	72.64	80.19
radt9-fhov727	226	65.04	94.25	96.46	97.35
radt9 scene	890	49.44	79.21	90.00	92.47
mmbld-74ov	88	77.27	79.55	80.68	85.23
mmbld-51ov	85	58.82	88.24	90.59	91.76
mmbld scene	173	68.21	83.82	84.97	88.44
avenches5882	96	6.25	31.25	48.96	60.42
avenches5874	102	8.82	17.65	24.51	33.33
avenches5873	91	15.38	34.07	48.35	59.34
avenches5883	102	9.80	25.49	45.10	54.90
avenchindust scene	391	9.97	26.85	41.43	51.66

Table 13: Summary of floor/peak detection performances based on distance

Image	npts	% ≤ 1 pixel	% ≤ 2 pixels	% ≤ 3 pixels	% ≤ 4 pixels
j1	1652	28.51	40.19	44.19	49.64
j2	1652	16.65	34.26	48.55	52.66
j3	1505	36.81	61.00	68.17	71.96
j4	1428	30.53	45.31	55.32	63.38
j5	1442	15.67	30.44	38.14	44.31
j6	1652	34.20	61.80	69.92	76.63
j7	1484	33.89	57.75	67.12	71.29
j8	1659	31.34	53.95	57.81	59.07
j scene	12474	28.46	48.16	56.18	61.10
radt9-fhov1627	678	34.66	71.68	85.69	90.12
radt9-fhov927	678	38.35	77.29	89.23	94.54
radt9-fhov727	678	43.36	81.86	92.18	95.72
radt9-fhov525	636	38.05	77.04	89.94	95.60
radt9 scene	2670	38.61	76.97	89.25	93.97
mmbld-74ov	88	22.73	44.32	54.55	80.68
mmbld-51ov	85	38.82	78.82	87.06	100.00
mmbld scene	173	30.64	61.27	70.52	90.17
avenches5882	288	7.99	14.58	25.35	32.99
avenches5874	306	2.61	8.17	15.69	21.90
avenches5873	273	6.96	20.51	28.21	34.80
avenches5883	306	3.92	10.13	28.76	35.95
avenchindust scene	1173	5.29	13.13	24.38	31.29

Table 14: Summary of epipolar matching performances based on distance



(a) Performance of The Floor/Peak Detection

(b) Performance of The Epipolar Matching Process

Figure 34: Performance evaluation of more test scenes

automated processes cannot produce edges and points at the expected location, even when the error is less than one pixel, most users will be compelled to fine-tune the automated solutions, reducing the usefulness of the automated processes.

Results in this section show the sensitivity and accuracy of the automated processes, and that these automated processes, while effective do not always perform perfectly in many situations. However, since these automated processes are designed for a semi-automated environment, performance evaluation within the context of SITECITY is required to determine the usefulness of these automated processes.

4.6 Performance Analysis of the Semi-Automated System

In the previous section, the performance of the automated processes without human intervention was evaluated. However, these evaluations do not determine the impact of the automated processes on the overall system. In this section, the usability of the automated processes in the semi-automated system is considered and the analysis is based on the discussion presented in Section 2. The usability of a semi-automated system is estimated by comparing the number of tasks executed and the elapsed time of the semi-automated system and the fully manual system. The elapse time includes the interaction, execution and waiting time. This comparison allows us to determine the effect of the automated processes in SITECITY. SITECITY is instrumented to record operations and elapsed time for every manual and automated process for the duration of the measurements. An example of the measurement record for measuring a peak roof building using the fully manual system is shown in Table 15. Table 16 shows the measurement record for the same task using the semi-automated system. The *Window* column shows the window of the operations. If an object was selected for manipulation, the name of the object is shown in the *Object* column. *Operation* and *Mode* columns define the start and the end of the state of an operation. Finally, the last column shows the time stamp at the start and end of each operation.

The complexity of these tasks can be reduced by collecting and summarizing tasks with similar functionalities. These tasks are broken down into 5 classes, Automated, Administrative, Measurement, System and Idle, as shown in Table 17. Automated tasks are the set of the automated processes described in Section 3. Administrative tasks are tasks not related to the actual image measurement, such as zooming or panning an image, or selecting a menu option. Measurement tasks require actual image measurements, such as measuring a pixel position to add a point or selecting an existing object to modify its position. System tasks include geometric constraints and triangulation. Finally, Idle tasks are tasks that cannot be directly measured, such as when a user attempts to find the correct menu option, deciding the next task or studying the image.

For this evaluation, the timing and operational statistics for the radt9 scene are not considered, because it was meant to be a test scene for subjects to become familiar with SITECITY. Table 18 shows the time and operation statistics for all measurements performed by all subjects. For each subject, the subject's group in the experiment (See Table 2) is shown, along with the scene and the method used, semi-automated (ap) or fully manual (m). In addition, for each task class, the number of operations performed and the amount of time used to perform those operations are displayed. The same information is plotted in Figure 35.

These plots exhibited large variations between users. In order to reduce the effects of user variations, the usability analysis of SITECITY is accomplished by considering two aspects of the measurement statistics. First, the total elapsed time and the number of executed tasks are considered. These two metrics represent the time and number of tasks needed to completely measure a scene regardless of the task classifications. The problem with using the total tasks and time statistic is that they can be easily influenced by user variations. Most of these variations are hidden in the idle time because we cannot directly observe a user's actions. Another class of tasks that can be influenced by user variations is the administrative tasks. While administrative tasks are performed directly by users, they are *non-essential* operations and more susceptible to user variation. The term *non-essential task* is used to denote a task that does not directly contribute to the measurement of a feature. For example, it is possible to measure a building without ever having to zoom or pan images. Contrast this to the measurement tasks, which are considered *essential*, because it is impossible to measure a building without performing measurement tasks.

One example illustrating the effects of user variations on administrative tasks concerns the user's preference on the viewing of images. Some users like to view objects at higher resolution, which requires more image manipulation because there is less of the visible image in a window. This user will have a larger

Window	Object	Operation	Mode	Real time
SiteCity		SiteCity	Start	0.917040
radt9ob.img		Zoom	Start	1.674016
radt9ob.img		Zoom	Exit	1.738432
radt9ob.img		Zoom	Start	2.152832
radt9ob.img		Zoom	Exit	2.367552
radt9ob.img		Zoom	Start	2.548112
radt9ob.img		Zoom	Exit	2.988288
radt9ob.img		Zoom	Start	3.335344
radt9ob.img		Zoom	Exit	3.672064
radt9.img		Zoom	Start	4.945344
radt9.img		Zoom	Exit	5.157712
radt9.img		Zoom	Start	5.512976
radt9.img		Zoom	Exit	5.772592
radt9.img		Zoom	Start	6.227008
radt9.img		Zoom	Exit	6.574464
radt9.img		Zoom	Start	6.924848
radt9.img		Zoom	Exit	7.437824
radt9.img		Pan Image	Start	7.857504
radt9.img		Pan Image	Exit	8.976576
radt9ob.img		Pan Image	Start	10.012688
radt9ob.img		Pan Image	Exit	10.641808
radt9ob.img		Create	State	16.147952
radt9ob.img		Peak Roof	State	18.925824
radt9ob.img		Measure	Start	20.906304
radt9ob.img		Measure	Exit	21.670112
radt9ob.img		Measure	Start	22.360720
radt9ob.img		Measure	Exit	22.450512
radt9ob.img		Measure	Start	23.548112
radt9ob.img		Measure	Exit	23.636928
radt9ob.img		Measure	Start	24.512976
radt9ob.img		Measure	Exit	24.604720
SiteCity		Finished	Start	25.533472
SiteCity	building	Finished	Exit	26.227984
radt9ob.img		Modify	State	28.048800
radt9ob.img		Line	State	31.064416
radt9ob.img		Modify	Start	32.189920
radt9ob.img	building		State	32.191872
radt9ob.img		Modify	Exit	33.763808
radt9ob.img		Polygon	State	37.850672

(a)

Window	Object	Operation	Mode	Real time
SiteCity		Finished	Start	37.850672
SiteCity	building	Finished	Exit	38.627168
radt9ob.img		Modify	Start	38.629120
radt9ob.img	building		State	38.630096
radt9ob.img		Modify	Exit	41.686704
radt9.img		Modify	Start	44.689632
radt9.img	building		State	44.690608
radt9.img		Modify	Exit	45.253360
radt9.img		Modify	State	47.422208
SiteCity		Finished	Start	47.422208
SiteCity	building	Finished	Exit	48.218224
radt9.img		Polygon	State	50.470032
radt9.img		Modify	Start	51.407568
radt9.img	building		State	51.408544
radt9.img		Modify	Exit	54.238720
radt9.img		Modify	Start	55.014640
radt9.img	building		State	55.017568
radt9.img		Modify	Exit	57.417328
radt9.img		Modify	Start	57.679872
radt9.img	building		State	57.681824
radt9.img		Modify	Exit	57.913136
radt9.img		Modify	Start	58.088816
radt9.img	building		State	58.091744
radt9.img		Modify	Exit	58.193824
radt9.img		Modify	Start	58.752096
radt9.img	building		State	58.754048
radt9.img		Modify	Exit	61.534448
radt9.img		Line	State	64.858480
SiteCity		Finished	Start	64.858480
SiteCity	building	Finished	Exit	65.733552
radt9.img		Modify	Start	65.827248
radt9.img	building		State	65.829200
radt9.img		Modify	Exit	68.116144
SiteCity		Constraints	Start	72.356816
SiteCity		Constraints	Exit	73.221152
SiteCity		Save	Start	81.356816
SiteCity		Save	Exit	81.458320
SiteCity		SiteCity	Exit	84.227008

(b)

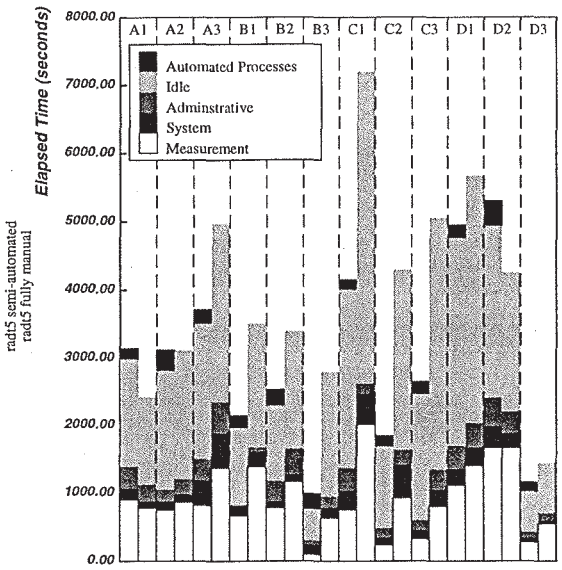
Table 15: Raw tasks and time output for manual measurement of a peak roof building in two images

Window	Object	Operation	Mode	Real time
SiteCity		SiteCity	Start	0.917040
radt9ob.img		Zoom	Start	1.494832
radt9ob.img		Zoom	Exit	1.599264
radt9ob.img		Zoom	Start	1.866288
radt9ob.img		Zoom	Exit	2.086864
radt9ob.img		Zoom	Start	2.441152
radt9ob.img		Zoom	Exit	2.715984
radt9ob.img		Zoom	Start	3.298656
radt9ob.img		Zoom	Exit	3.687680
radt9.img		Zoom	Start	4.312320
radt9.img		Zoom	Exit	4.489952
radt9.img		Zoom	Start	4.762832
radt9.img		Zoom	Exit	5.018544
radt9.img		Zoom	Start	5.345504
radt9.img		Zoom	Exit	5.694512
radt9.img		Zoom	Start	6.257664
radt9.img		Zoom	Exit	6.771616
radt9.img		Pan Image	Start	7.050752
radt9.img		Pan Image	Exit	7.463600
radt9.img		Create	State	19.669136
radt9.img		Peak Roof	State	22.711104
radt9.img		Measure	Start	24.183488
radt9.img		Measure	Exit	25.674016
radt9.img		Measure	Start	25.978528
radt9.img		Measure	Exit	26.067344
radt9.img		Measure	Start	27.319152
radt9.img		Measure	Exit	27.411872
radt9.img		Measure	Start	28.372832
radt9.img		Measure	Exit	28.491904
SiteCity		Finished	Start	29.456768
SiteCity	Building	Auto Peak	Start	29.997072
SiteCity	Building	Auto Peak	Exit	31.407968
SiteCity	Building	Auto Floor	Start	31.407968
SiteCity		Auto Copy	Start	52.392352
SiteCity	Building	Auto Copy	Exit	54.868240
SiteCity	Building	Auto Floor	Exit	54.985360
SiteCity	Building	Auto Localization	Start	55.000976
SiteCity	Building	Auto Localization	Exit	56.040016
SiteCity	Building	Finished	Exit	56.858480
SiteCity		Save	Start	61.096624
SiteCity		Save	Exit	61.211792
SiteCity		SiteCity	Exit	63.640832

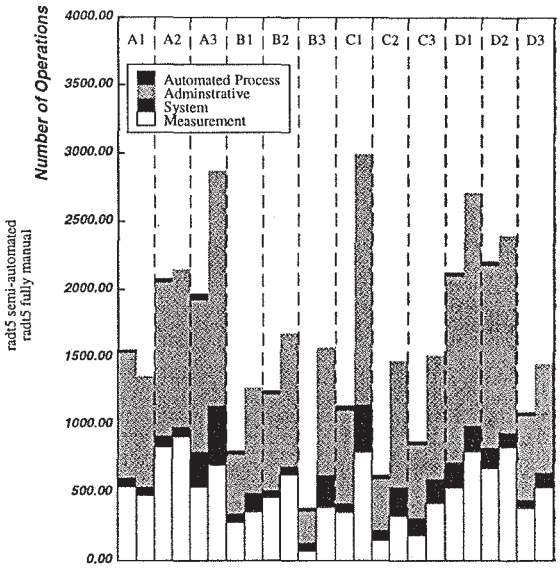
Table 16: Raw tasks and time output for semi-automated measurements of a peak roof building in two images

Task Classes	Automated Processes	Administrative	Measurement	System	Idle
Tasks	Auto Peak Auto Floor Auto Copy Auto Localization	Zoom Image Pan Image	Measure Modify	Constraints Save	

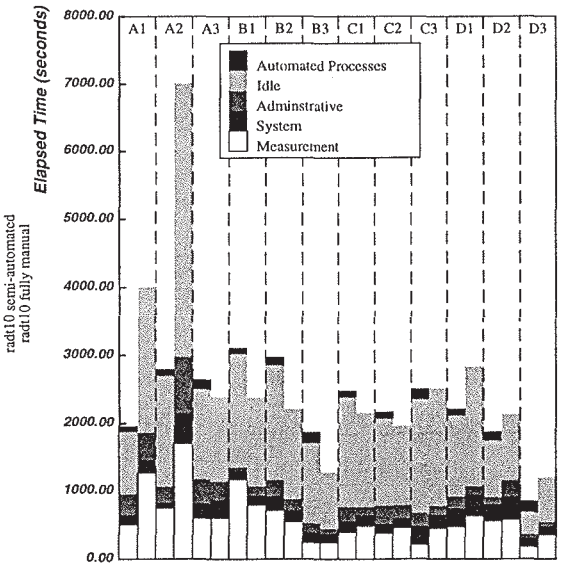
Table 17: Classes of operation and tasks



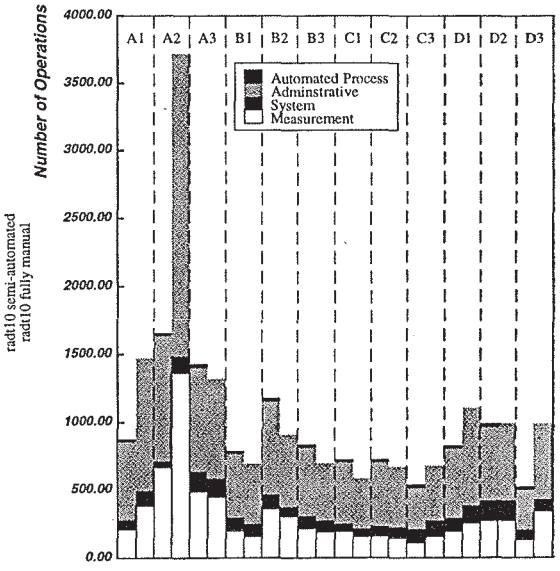
(a) Histogram of the elapsed time for the radt5 scene



(b) Histogram of the number of operations for the radt5 scene



(c) Histogram of the elapsed time for the radt10 scene



(d) Histogram of the number of operations for the radt10 scene

Figure 35: Plot of elapsed time and number of operations for all measurements

Subject	Scene-Method	Automated	Measurement	Administrative	System	Idle	Total
A1	radt5-ap	164.26(19)	897.22(542)	164.38(68)	337.00(927)	1583.80	3146.66(1556)
A1	radt5-m	0.00(0)	775.43(477)	97.71(67)	256.71(818)	1290.35	2420.19(1362)
A1	radt10-ap	77.93(16)	502.35(209)	144.05(71)	307.50(579)	925.24	1957.08(875)
A1	radt10-m	0.00(0)	1262.74(383)	184.98(111)	418.64(976)	2134.15	4000.52(1470)
A2	radt5-ap	310.94(29)	748.76(837)	125.32(82)	184.75(1135)	1751.75	3121.51(2083)
A2	radt5-m	0.00(0)	860.91(912)	121.59(72)	225.53(1165)	1897.85	3105.88(2149)
A2	radt10-ap	95.70(19)	741.29(667)	79.15(45)	242.95(923)	1631.94	2791.03(1654)
A2	radt10-m	0.00(0)	1703.21(1362)	451.50(122)	818.18(2237)	4028.90	7001.79(3721)
A3	radt5-ap	214.26(45)	813.93(539)	373.67(263)	317.88(1123)	1992.96	3712.70(1970)
A3	radt5-m	0.00(0)	1367.61(702)	518.88(440)	458.99(1730)	2616.15	4961.62(2872)
A3	radt10-ap	141.97(26)	597.80(487)	219.62(149)	360.20(767)	1326.46	2646.04(1429)
A3	radt10-m	0.00(0)	598.40(447)	268.91(139)	277.59(736)	1245.61	2390.52(1322)
B1	radt5-ap	185.99(27)	660.97(282)	147.79(66)	23.29(433)	1128.57	2146.61(808)
B1	radt5-m	0.00(0)	1386.67(357)	219.83(141)	65.88(780)	1830.41	3502.79(1278)
B1	radt10-ap	81.63(17)	1168.09(197)	125.64(97)	55.12(478)	1674.98	3105.46(789)
B1	radt10-m	0.00(0)	790.57(155)	141.09(93)	139.05(446)	1302.36	2373.07(694)
B2	radt5-ap	246.53(29)	782.07(463)	89.87(51)	304.92(713)	1112.69	2536.09(1256)
B2	radt5-m	0.00(0)	1162.80(630)	106.52(67)	384.63(984)	1734.21	3388.16(1681)
B2	radt10-ap	119.47(23)	709.86(360)	217.48(105)	238.30(690)	1689.36	2974.47(1178)
B2	radt10-m	0.00(0)	540.29(303)	179.73(72)	162.48(537)	1333.30	2215.80(912)
B3	radt5-ap	230.34(26)	96.70(70)	138.60(63)	70.52(228)	460.91	997.07(387)
B3	radt5-m	0.00(0)	628.26(390)	158.78(238)	163.77(947)	1842.85	2793.66(1575)
B3	radt10-ap	161.34(20)	236.29(213)	155.76(95)	136.28(507)	1181.10	1870.78(835)
B3	radt10-m	0.00(0)	237.32(190)	130.46(85)	75.93(421)	833.45	1277.16(696)
C1	radt5-ap	148.65(35)	753.57(352)	277.31(69)	338.03(687)	2635.51	4153.07(1143)
C1	radt5-m	0.00(0)	2005.99(796)	463.93(356)	157.10(1841)	4564.02	7191.04(2993)
C1	radt10-ap	93.96(20)	394.40(191)	156.88(56)	217.61(458)	1614.72	2477.58(725)
C1	radt10-m	0.00(0)	478.74(154)	164.12(61)	127.39(371)	1388.26	2158.51(586)
C2	radt5-ap	158.32(32)	232.38(147)	110.85(80)	149.70(370)	1194.09	1845.34(629)
C2	radt5-m	0.00(0)	923.71(328)	488.09(212)	222.01(938)	2651.45	4285.25(1478)
C2	radt10-ap	93.59(18)	375.66(156)	135.12(78)	272.13(473)	1292.22	2168.71(725)
C2	radt10-m	0.00(0)	453.10(142)	140.93(85)	200.68(447)	1163.56	1958.27(674)
C3	radt5-ap	187.78(26)	323.04(184)	131.68(129)	148.78(540)	1858.42	2649.70(879)
C3	radt5-m	0.00(0)	797.58(421)	247.02(182)	299.67(912)	3708.30	5052.57(1515)
C3	radt10-ap	152.08(25)	206.85(107)	274.56(103)	198.18(304)	1672.79	2504.46(539)
C3	radt10-m	0.00(0)	444.29(155)	204.72(119)	130.87(403)	1729.51	2509.39(677)
D1	radt5-ap	190.74(25)	1103.93(535)	241.48(188)	353.31(1376)	3063.32	4952.78(2124)
D1	radt5-m	0.00(0)	1399.20(800)	267.61(194)	361.91(1714)	3640.95	5669.67(2708)
D1	radt10-ap	95.12(19)	474.88(192)	261.85(100)	181.29(517)	1197.91	2211.05(828)
D1	radt10-m	0.00(0)	626.88(254)	316.00(131)	123.71(722)	1768.55	2835.15(1107)
D2	radt5-ap	367.14(31)	1664.81(675)	318.54(157)	436.48(1345)	2521.29	5308.26(2208)
D2	radt5-m	0.00(0)	1665.22(835)	219.48(110)	324.70(1451)	2045.91	4255.31(2396)
D2	radt10-ap	129.79(24)	554.37(269)	248.59(154)	107.23(544)	836.70	1876.68(991)
D2	radt10-m	0.00(0)	578.57(273)	344.24(152)	235.50(571)	976.45	2134.76(996)
D3	radt5-ap	143.21(26)	274.81(382)	65.13(69)	92.15(618)	588.19	1163.49(1095)
D3	radt5-m	0.00(0)	544.24(534)	43.29(113)	117.24(808)	724.23	1429.00(1455)
D3	radt10-ap	162.76(21)	175.86(130)	128.58(79)	57.79(292)	329.54	854.52(522)
D3	radt10-m	0.00(0)	348.38(340)	128.84(93)	70.56(557)	649.99	1197.77(990)

Table 18: Tasks and elapse time break down for all measurements

number of administrative tasks. In addition, a user more proficient with SITECITY will also have fewer administrative tasks because he/she knows how to use SITECITY efficiently. For example, a proficient user can set up windows and images and perform his/her measurements in such a way that he/she only needs to perform a minimum number of image manipulations. Another way a proficient user can make a difference is to perform all measurement tasks in one image to minimize the number of changes in the selected operations and objects, reducing number of menu selection tasks.

Therefore, in order to reduce the effects of user variations on the analysis, we define *user tasks* and *user time* to include only essential operations of the system 19. The number of user tasks is defined to be the number of measurement tasks and the user time is defined to be the sum of the elapsed time for the automated tasks and the measurement tasks. System time and tasks are not included because those tasks are not performed by users. Measurement tasks are *essential* tasks that all users have to perform regardless of proficiency and other variations. Automated process time is included in the user time because it is performing some tasks for users. In this scheme, we define the *user cost* of SITECITY to be the number of *user tasks*, and the unit tasks of SITECITY are the tasks of measuring or modifying the position of an object.

However, the amount of administrative, system, and idle tasks can be influenced by the *user cost* of SITECITY. If a user spends less time measuring, he/she will perform fewer administrative tasks to support his/her measurement tasks. Therefore, both user time/tasks and total time/tasks are considered in the evaluation of the semi-automated system.

Essential Tasks	Non-Essential Tasks
Automated Tasks	Idle
Measurement Tasks	Administrative Tasks
	System Tasks

Table 19: Essential and non-essential tasks

Table 20 presents the average elapsed time and number of tasks for both the semi-automated and fully manual measurements. The first column shows the scene and the methods; *radt5-ap* denotes the semi-automated measurement statistics gathered for the radt5 scene, whereas *radt5-m* denotes statistics gathered for the radt5 scene using the fully manual system. This table shows the average elapsed time and the average number of operations for a user to completely measure a scene using a particular method. Out of 24 pairs of semi-automated and fully manual measurements, there are only 9 cases where the semi-automated measurement takes more time and effort than the fully manual measurement, and 6 of the 9 cases involve the radt10 scene. This is expected after the evaluation of the automated processes presented in Section 4.5.2, where the automated processes performed poorer in the radt10 scene than the radt5 scene.

Method	Sample Size	System	Administrative	Idle	Total	User
radt5-ap	12	182.05(107.08)	229.73(791.25)	1657.62	2977.77(1344.83)	908.36(417.33)
radt5-m	12	246.06(182.67)	253.18(1174.00)	2378.89	4004.60(1955.17)	1126.47(598.50)
radt10-ap	12	178.94(94.33)	197.88(544.33)	1281.08	2286.49(924.17)	628.59(264.83)
radt10-m	12	221.29(105.25)	231.72(702.00)	1546.17	2671.06(1153.75)	671.87(346.50)

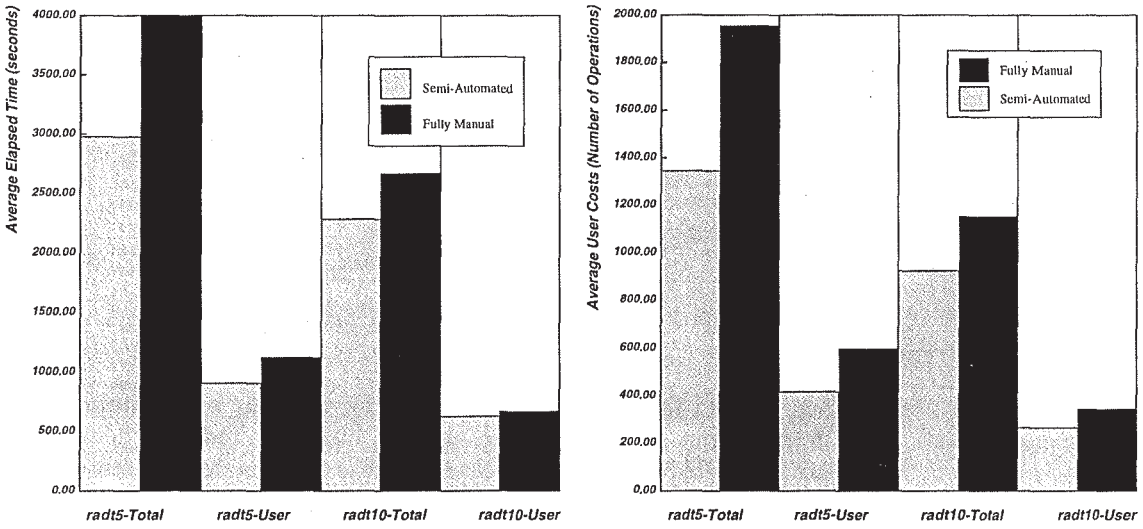
Table 20: Overall measurement statistics — average time(tasks)

The average user and total elapsed time/number of tasks is plotted in Figure 36. The results show that there are significant improvements in terms of elapsed time and number of performed tasks when the semi-automated system is used on the radt5 scene. Directly comparing the average elapsed time/tasks between the semi-automated and the fully manual methods for each scene reveals that, overall, a subject spent about 20% less time and tasks than his/her counterpart when using the semi-automated system. The semi-automated system also seems to reduce the variation of human performance in terms of both the elapsed time and

Method	Sample Size	System	Administrative	Idle	Total	User
radt5-ap	12	97.38(64.95)	133.67(384.76)	809.76	1374.88(630.91)	476.26(225.79)
radt5-m	12	161.82(117.24)	117.27(398.90)	1109.23	1570.51(633.75)	446.77(206.91)
radt10-ap	12	62.89(32.66)	95.45(179.88)	421.94	612.84(341.62)	260.41(164.11)
radt10-m	12	102.53(28.08)	208.52(514.05)	883.94	1541.03(854.51)	414.59(335.11)

Table 21: Overall measurement statistics — standard deviation time (tasks)

number of tasks. Table 21 shows the standard deviation values for the measured elapsed time and tasks for each scene using different methods. The standard deviation values for the semi-automated system are consistently smaller than the fully manual system. One explanation is that the subjects are performing fewer operations.



(a) User/total elapsed time

(b) User/total tasks

Figure 36: Usability analysis on radt5 and radt10 scene

4.6.1 Analysis of the Effects of the Order of the Measurements

The previous analysis considers the overall effect of the automated processes. Another way to analyze the effect of the automated processes is to consider how they affect the performances for individual subjects. Since each subject measures the same scene twice, we can measure the performance of the second measurement of a scene compared to the first measurement of the same scene. Subjects are expected to measure a scene faster and with fewer number of operations the second time, because they are more familiar with the scene and more proficient with the interface.

The differences between the first and the second set of measurements for the same scene is computed for each subject. Let $S = \{s_1, \ldots, s_m\}$ be a set of scenes, and $U = \{u_1, \ldots, u_n\}$ be a set of users. If $a_{s,u}$ is the statistics from the first set of measurements on scene s measured by subject u, and $b_{s,u}$ is the second set, then the difference in the elapsed time between the first and second set of measurements is

$$d_t(a_{s,u}, b_{s,u}) = t(a_{s,u}) - t(b_{s,u}),$$

where $t(a)$ returns the elapsed time for measurement set a. Likewise, the differences in terms of the number of operations is defined as

$$d_T(a_{s,u}, b_{s,u}) = T(a_{s,u}) - T(b_{s,u}),$$

where $T(a)$ returns the number of operations for measurement set a.

Based on the subject groups presented in Table 2, $a_{s,u}$ and $b_{s,u}$ cannot be measured with the same method for the same s and u. We write $a^{ap}_{s,u}$ to denote the first set of measurements performed by user u on scene s using the semi-automated approach, and $b^{m}_{s,u}$ to denote the second set of measurements performed by user u on scene s using the fully manual approach.

Let

$$D^{ap}_t = \{d_t(a^m_{s_i,u_j}, b^{ap}_{s_i,u_j})\} \; \forall \; s_i \in S, u_j \in U$$

be a set of differences of between the first and second set of measurements where the first set of measurements was performed using the fully manual method and the second set used the semi-automated method. Likewise, we define

$$D^{ap}_T = \{d_T(a^m_{s_i,u_j}, b^{ap}_{s_i,u_j})\} \; \forall \; s_i \in S, u_j \in U,$$

$$D^{m}_t = \{d_t(a^{ap}_{s_i,u_j}, b^{m}_{s_i,u_j})\} \; \forall \; s_i \in S, u_j \in U,$$

and

$$D^{m}_T = \{d_T(a^{ap}_{s_i,u_j}, b^{m}_{s_i,u_j})\} \; \forall \; s_i \in S, u_j \in U.$$

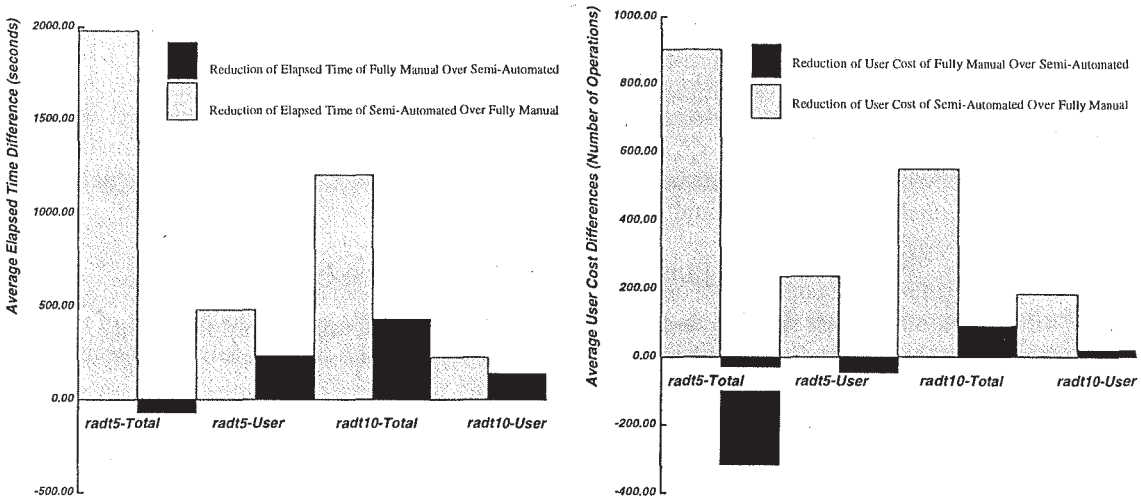
The average values of D^{ap}_t,D^{ap}_T,D^{m}_t and D^{m}_T are presented in Table 22 and Figure 37. A positive value means that it takes less time and/or fewer tasks to perform the measurement task in the second set of measurements. The standard deviation values for D^{ap}_t,D^{ap}_T,D^{m}_t and D^{m}_T are shown in Table 23. When the second method is the semi-automated system, the improvement to both the elapsed time and number of tasks performed is dramatically better for an average user. For the radt5 scene, the average user actually performed more tasks using the fully manual system even when he/she has measured the same scene before using the semi-automated system. The improvements are more significant for the radt5 scene than radt10, because the performance evaluation of the automated processes presented in Section 4.5.2 reveals that the automated processes performed better in the radt5 scene. An additional explanation might be that there are a number of similar buildings in radt5, and that the scene complexity is lower for the radt5 scene, due to fewer occlusions.

1st Method	2nd Method	Sample Size	System	Administrative	Idle	Total	User
radt5-m	radt5-ap	6	131.3(123.0)	43.0(571.8)	1323.5	1980.9(903.0)	483.1(237.3)
radt5-ap	radt5-m	6	3.3(−28.2)	−3.9(−193.7)	−119.0	−72.7(−317.7)	46.9(−125.0)
radt10-m	radt10-ap	6	102.1(25.0)	114.5(362.8)	759.3	1204.0(551.2)	228.1(184.2)
radt10-ap	radt10-m	6	17.4(3.2)	46.9(47.5)	229.1	434.9(92.0)	141.5(20.8)

Table 22: Differences between the first and the second set of measurements — average time(tasks)

1st Method	2nd Method	Sample Size	System	Administrative	Idle	Total	User
radt5-m	radt5-ap	6	136.24(98.32)	115.34(329.11)	555.64	800.58(541.89)	341.84(129.63)
radt5-ap	radt5-m	6	85.38(78.55)	90.69(252.10)	444.74	865.93(388.88)	277.38(110.99)
radt10-m	radt10-ap	6	135.85(31.75)	241.36(491.58)	916.73	1664.18(788.94)	433.24(268.09)
radt10-ap	radt10-m	6	33.02(17.17)	64.85(85.25)	168.85	308.47(133.63)	203.77(36.90)

Table 23: Differences between the first and the second set of measurements — standard deviation time(tasks)



(a) Differences in time

(b) Differences in number of operations

Figure 37: Differences between the first and the second set of measurements

4.6.2 Analysis of the Composition of User Tasks

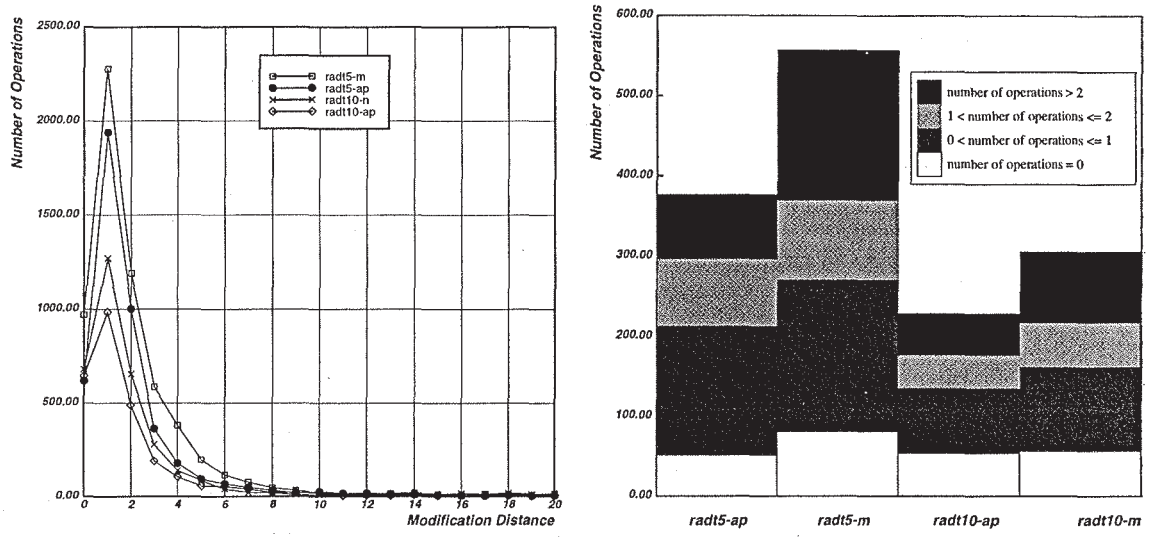
The final analysis is to determine the composition of the user tasks. In our scheme, user tasks are broken down into 3 classes; modifications of a measured object, measurements of a new point, and deletions. These modifications are further divided according to the distance over which an object is modified. Table 24 shows the decomposition of the user tasks for the average and standard deviation values. The majority of the user tasks are modification tasks, which means that the user spends most of his/her time moving points around. This is consistent with the user interface of SITECITY. Recall from Section 3 that a building is measured by measuring the roof points in one image, followed by modifications of the corner points of the hypothesized building object in all images. Therefore, there should be more modification tasks than measurement tasks. The number of delete tasks for the fully manual method is slightly higher than the semi-automated method, which is the reason that the number of measurement tasks are higher for the fully manual approaches. When a user makes a mistake in a measurement and deletes it, he/she will have to perform the measurement again. The standard deviation values show that there is less variation when the semi-automated system is used.

Method	Sample Size	User Tasks	Modification Tasks	Delete Tasks	Measure Tasks
radt5-ap	12	417.33(225.79)	376.92(223.70)	1.92(2.47)	23.50(13.90)
radt5-m	12	598.50(206.91)	557.25(204.70)	2.50(3.92)	31.17(21.36)
radt10-ap	12	264.83(164.11)	229.00(164.46)	0.25(0.62)	34.08(3.23)
radt10-m	12	346.50(335.11)	306.08(329.16)	0.92(1.16)	39.42(8.00)

Table 24: Breakdown of user tasks — average (standard deviation)

Since modification tasks are the bulk of the user tasks, the composition of the modification tasks needs to be analyzed. Figure 38(a) shows a histogram of all modification tasks based on the modified distances. The graph is truncated where the modification distance is greater than 20 pixels because the plot exhibits the same asymptotic trend. Table 25 shows a coarse breakdown of the modification tasks in terms of translated distance in pixels. The same information is also plotted in Figures 38(b). The figure and the table show the average and standard deviation of the number of modification operations that are either translated by a

distance of zero pixels, less than and equal to one pixel, and so on. A modification of zero pixels means that an object was selected, but not changed. Selection of objects accounts for about 17% of all the modification tasks performed. The majority of modifications, about 40%, have distance of less than one pixel. This suggests that the users spend most of the time fine-tuning edge and corner positions. The average number of operations where the modified distance is less than two pixels is similar for both methods. It appears that approximately the same amount of work is spent fine-tuning the measurements regardless of the method used. The benefit of the automated processes appears to be the reduction of modification tasks whose distance is greater than two pixels.



(a) Break down of modification tasks

(b) Average number of operations per scene measurement

Figure 38: Breakdown of modification tasks in terms of translation distances

Method	Size	= 0 pixel	≤ 1 pixel	≤ 2 pixels	> 2 pixels
radt5-ap	12	51.58(46.49)	161.58(109.67)	83.50(52.32)	80.25(50.98)
radt5-m	12	80.92(73.84)	189.75(131.00)	99.17(48.87)	187.42(68.64)
radt10-ap	12	53.83(49.10)	82.25(78.15)	40.67(28.81)	52.25(27.25)
radt10-n	12	56.75(67.57)	105.92(129.35)	54.58(73.66)	88.83(73.23)

Table 25: Breakdown of modification tasks in terms of translation distances — average (standard deviation)

4.6.3 Summary of Performance Analysis of the Semi-Automated System

It is difficult to conclusively demonstrate the *usability* of SITECITY with the experiments shown in this paper due to small number of subjects and scenes. However, the results are promising. On average, automated processes reduced the elapsed time by 20% and *user cost* by 12%. In addition, when users are familiar with the scene, the automated processes offer greater assistance to users. The large standard deviation in Table 21 shows that there are large variations between users, which is understandable because all subjects were novices to SITECITY. However, there appears to be less variation when the semi-automated system is used. In future studies, it would be interesting to perform this experiment with expert users. Table 25 shows

the need for automated processes to have the ability to locate the corners and edges to sub-pixel accuracies. This should decrease the number of minute modifications.

Furthermore, informal feedbacks from the subjects reveals some interesting facts. First, most subjects think the automated processes do help in their tasks and they felt that they do less work in the semi-automated approach. When given a choice of using the semi-automated system and the fully-manual system, most of them choose the semi-automated system. However, some would have preferred the ability to control the activation of the automated processes, because they found it performed better when the buildings and images have good contrast and they would use the automated processes on those buildings and images. On images and buildings where they think the automated processes would fail, they preferred to do the measurements manually. The most surprising response is that some of our subjects prefer the waiting in the semi-automated system. They said that waiting for the automated processes gives them a moment to think about what they are doing, and makes them feel less rushed. One possible explanation for this respond can be attributed entirely to subjects not having to perform these tasks under pressures or deadlines, even thought they were told to perform the tasks with accuracy and speed.

It was previously assumed that lag in an interactive system is detrimental to human performance. In [34], the lag was shown to reduce the usability of an interactive system for motor-sensory tasks, such as a system for virtual reality application. However, as the previous anecdotal evidence suggests, it is possible that some forms of human-computer interaction, where immediate feedback is not as crucial, benefit from an appropriate amount of lag can enhance the productivity of an user. In effect, this lag forces users to relax and take a break between measurement tasks. This possibility suggests a future study to determine the effect of the speed of the automated processes on system performance in the context of site modeling tasks.

4.6.4 Productivity of the Semi-Automated System

The usability analysis of SITECITY presented in Section 4.6 demonstrated the *usability* of semi-automated SITECITY in comparison with the fully manual version. We found that the semi-automated version of SITECITY is more usable than the fully manual version. However, this analysis did not address the *productivity* of SITECITY, which measures the *usability* of SITECITY in a production environment. A system that is not *productive* will not be used, because it will not be cost effective to use the system. The *productivity* of a system can be defined as the amount of work required to achieve a unit of *benefit*. A system is more *productive* if it requires less work to achieve the same amount of *benefit* as its less *productive* counterpart.

For the site modeling task, the *benefit* of a system can be defined as a function of the *number of point measurements* and the *accuracy* of these measurements. The definition of *user cost* presented in Section 2.1 can be used to quantify the amount of work for the productivity analysis. In general, we hypothesize that site modeling systems tend to have operating cost curves similar to those shown in Figure 39, which shows the family of operating cost curves for a given system. Each line, $N = s(e)W$, relates the amount of work(W) required to perform N number of point measurements within a given error e, where s is the slope of the line, which is a function of e.

In this scheme, we assume that the same amount of work is required to perform a point measurement with a given accuracy for experienced operators. This model does not account for training cost. In one extreme, where $e = \textit{infinity}$, the cost curve tells us that very little work is needed to measure a lot of points with zero accuracy. On the other extreme, the $e = 0$ curve shows that a lot of work is required to measure few points that have no error. A more realistic operating region is somewhere in between these two extremes.

To determine if a system is *productive* for a given project, it is necessary to determine how many measurements needs to be performed (Nd); how much work/time a user can spend on the project (Wm); and the allowable error of these measurements (e_d). For the system with the operating cost curves and the desired performance of Wm and Nd shown in Figure 39, the system cannot be productive if users are required to achieve measurement errors of less than γ. For example, to measure Nd measurements whose measurement errors are within β, then the operating cost curves show that Wb units of work are required. Since the maximum amount work allowed is Wm, and $Wb > Wm$, the system is not *productive* for operating at this level of performance.

In Section 4.6, we evaluated the amount of work for a given number of measurements and a specified accuracy for these measurements. The results of these evaluation can be interpreted in terms of operating cost curves. In these evaluations, the same number of building points were measured using the semi-automated

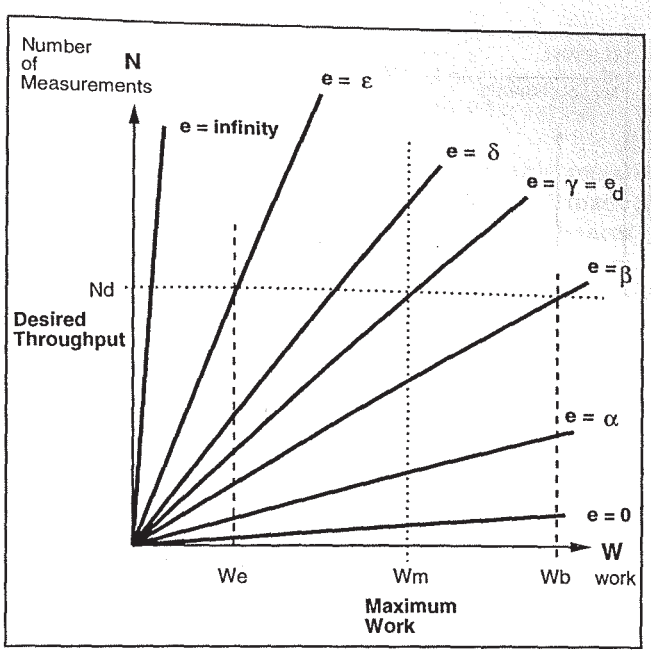


Figure 39: A typical operating cost curves for model construction systems

and the fully manual system (Section 4.2). Since Section 4.4 shows that the measurements between the semi-automated and fully manual SITECITY system are not significantly different for an average user, we concluded that these measurements have approximately the same error tolerance. According to the results presented in Table 20, *W0* (e.g., 1345 unit tasks for radt5 scene) is less than *W1* (e.g., 1955 unit tasks for radt5 scene) for an average user, as shown in Figure 40. Whether this is productive for a given site modeling project (i.e., $W0 \leq Wm$) depends on the project requirements. However, it is clear that the semi-automated approach is more productive than its fully manual counterpart.

5 Conclusions

In this paper, methods and criteria for designing and integrating automated processes in an interactive environment are presented. SITECITY uses these methods to integrate image understanding algorithms into an interactive system for extracting three dimensional building models using multiple images. These automated algorithms are unobtrusively integrated and reduce the complexity of the measurement tasks. SITECITY allows the user to measure building points on multiple images using a simple graphical interface. These building points are triangulated to estimate the three-dimensional position and precision, and constrained according to the geometric model. Complex building objects can be constructed by applying geometric constraints on several primitive components. The current system allows us to measure a large variety of building types. However, more general building object models are needed to handle scenes where the roof structures are complex. We are currently studying methods to construct other difficult building objects within SITECITY, such as the use of generic peak roof building type, and building models with overhanging roofs.

In addition, methods were presented to evaluate the performance of the semi-automated system. Most subjects perform the measurements faster and using fewer operations when using the semi-automated system. This reduction in the amount of work is not from sloppy measurements. The average difference between the semi-automated and fully manual measurements is less than 0.6 pixels. However, it is clear that more effort is needed to improve the automated processes. The data collected from our evaluations allows us to determine weaknesses in the system, and suggest some ideas for future improvements:

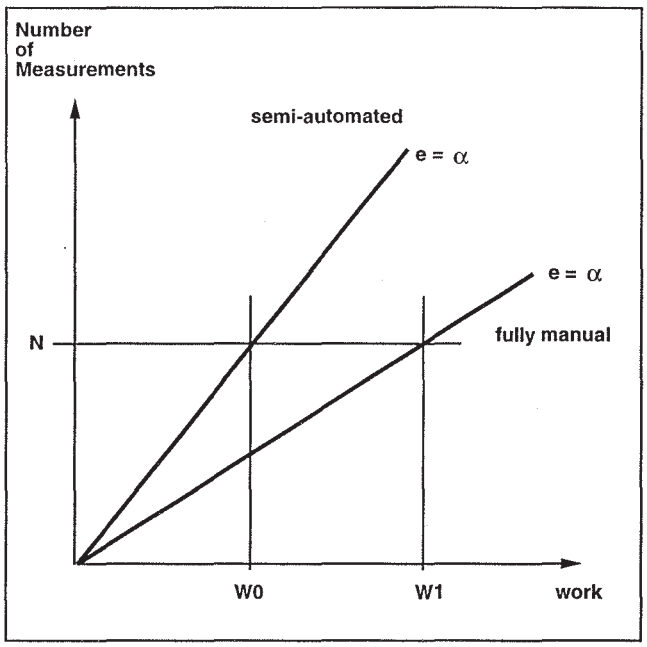


Figure 40: System cost curves for fully manual and semi-automated version of SITECITY

- Increased robustness in the automated processes should enhance the usability of the system.
- The ability to locate points more accurately in images should reduce the number of modification tasks, and make SITECITY more productive.
- A better understanding of idle time since it appears to account for more than 50% of all site modeling tasks. This may reflect user interface or task analysis problems.
- The ability to handle complex buildings will allow SITECITY to perform better on a variety of images and scenes.

In future work, we hope to combine the use of fully automated systems with SITECITY to create a comprehensive environment for building delineation, and to extend the concepts presented in this paper to other semi-automated feature delineation tasks, such as road extraction, stereo matching, and multispectral classification. At present, SITECITY is actively being used to generate accurate three dimensional ground truth for our automated building extraction research, and to populate our spatial databases with buildings for research in simulation of virtual worlds.

We have formulated principles for designing and evaluating semi-automated feature extraction systems. These principles are embodied in SITECITY, a semi-automated three-dimensional building delineation system. The rigorous evaluation of SITECITY presented in this paper shows the usability of the system and demonstrates that the application of the presented methods and criteria leads to a usable interactive vision system. The methods and criteria introduced in this paper leads naturally to the design of usable semi-automated vision systems in other domains such as road extraction, stereo matching, and multispectral classification.

The demand for rapid compilation of spatial data derived from remotely sensed imagery cannot be met by current state-of-the-art fully automated vision algorithms alone. However, the integration of fully automated algorithms into an interactive system, as illustrated in this paper, is a promising path for increasing the utility and productivity of computer vision techniques in traditionally tedious manual tasks.

6 Acknowledgements

This work would not have been possible without the support of the members of the Digital Mapping Laboratory. First, I would like to thank all of the members of the lab who agreed to spend 5 hours of their lives to become part of the statistics in the evaluation of SITECITY.

I would like to thank several people for their contributions to this research effort. Chris McGlone, godfather of SITECITY, provided the photogrammetric infrastructure that made SITECITY possible. He also developed and implemented the concept of geometric constraints which enhances the usability of the system, and coined the name, SITECITY, for the semi-automated site modeling system. Jeff Shufelt served as the sounding board for my ideas; his comments helped to focus the research direction of the work. I would also like to acknowledge Stephen Gifford and Chris Olson who provided the Image Display Library (IDL), which is the foundation of the user interface in SITECITY.

Chris McGlone, Steven Cochran, Jeff Shufelt, and Dave McKeown provided many useful comments on earlier drafts of this paper, in particular, Jeff Shufelt's comments and insights dramatically enhanced the clarity and technical content of this paper.

Finally, I would like thank Dave McKeown for his constant encouragement and support, and whose vision and direction made the Digital Mapping Laboratory an enjoyable and productive research environment.

References

[1] Z. Aviad. Locating corners in noisy curves by delineating imperfect sequences. Technical Report CMU–CS–88–199, Carnegie Mellon University, December 1988.

[2] Z. Aviad and P. D. Carnine. Road finding for road network extraction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 814–819, Ann Arbor, Michigan, June 5–9 1988.

[3] H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *Proc. IJCAI'81*, 1981.

[4] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[5] Stephen T. Barnard and Martin A. Fischler. Computational stereo. *ACM Computing Surveys*, 14(4):553–572, December 1982.

[6] P. R. Boniface. State-of-the-art in softcopy photogrammetry. In *Mapping and Remote Sensing Tools for the 21st Century*, pages 205–210, Washington, D. C., August 1994. ASPRS.

[7] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[8] Gunilla Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):849–865, November 1988.

[9] S. K. Card. Human factors and artificial intelligence. In *Intelligent Interfaces, Theory, Research and Design*, pages 27–41. Elsevier Science Publishers B.V., Amsterdam, 1989.

[10] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.

[11] A. Chapanis. Evaluating usability. In B. Shackel and S.J. Richardson, editors, *Human Factors for Informatics Usability*, pages 359–395. Cambridge University Press, Cambridge, 1991.

[12] Y. Cheng, R. Collins, A. Hanson, and E. Riseman. Model matching and extension for automated 3D site modeling. In *Proceedings of the DARPA Image Understanding Workshop*, pages 197–204, Washington, D. C., April 19–21 1993. Morgan Kaufmann Publishers, Inc.

[13] S. D. Cochran and G. Medioni. 3-D surface description from binocular stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):981–994, October 1992.

[14] Steven Douglas Cochran. Adaptive vergence for stereo matching. In *International Archives of Photogrammetry and Remote Sensing: Spatial Information from Digital Photogrammetry and Computer Vision*, volume 30, 3/1, pages 144–151, Munich, Germany, September 5–9 1994. Commission III.

[15] L. Drisis. A CAD-user survey: Experience and results. In P. Brödner and W. Karwowski, editors, *Ergonomics of Hybrid Automated Systems III*, pages 313–318. Elsevier Science Publishers B.V., 1992.

[16] W. Förstner. On the geometric precision of digital correlation. *Proceedings of ISPRS Commission III Symposium*, XXV:176–189, 1982.

[17] P. Fua and A. J. Hanson. Resegmentation using generic shape: Locating general cultural objects. Technical report, Artificial Intelligence Center, SRI International, May 1986.

[18] P. V. Fua and A. J. Hanson. Objective functions for feature discrimination: Applications to semi-automated and automated feature extraction. In *Proceedings of the DARPA Image Understanding Workshop*, Palo Alto, California, May 1989. Defense Advanced Research Projects Agency.

[19] S. J. Gee and A. M. Newman. Conceptual design for a model-supported exploitation workstation for imagery analysts. In E. B. Barrett and D. M McKeown, editors, *Integrating Photogrammetric Techniques with Scene Analysus and Machine Vision*, volume 1944, pages 242–253, Orlando, Florida, April 1993. SPIE.

[20] E. Gülch. Fundamentals of softcopy photogrammetric workstations. In *Mapping and Remote Sensing Tools for the 21st Century*, pages 193–204, Washington, D. C., August 1994. ASPRS.

[21] I. Hamburg. Interactive calculation during the design phase. In P. Brödner and W. Karwowski, editors, *Ergonomics of Hybrid Automated Systems III*, pages 325–330. Elsevier Science Publishers B.V., 1992.

[22] A. J. Hanson and L. H. Quam. Overview of the SRI cartographic modeling environment. In *Proceedings of the DARPA Image Understanding Workshop*, pages 576–582, Cambridge, Massachusetts, April 1988. Defense Advanced Research Projects Agency.

[23] D. I. Havelock. Geometric precision in noise-free digital images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1065–1075, October 1989.

[24] C. Heipke. State-of-the-art of digital photogrammetric workstations for topographic applications. *Photogrammetric Engineering and Remote Sensing*, 61(1):49–56, January 1995.

[25] A. Huertas, W. Cole, and R. Nevatia. Using generic knowledge in analysis of aerial scenes: A case study. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1642–1648, Detroit, Michigan, August 1989.

[26] A. Huertas and R. Nevatia. Detecting buildings in aerial images. *Computer Vision, Graphics, and Image Processing*, 41(2):131–152, February 1988.

[27] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

[28] R. B. Irvin and D. M. McKeown. Methods for exploiting the relationship between buildings and their shadows in aerial imagery. *IEEE Transactions on Systems, Man & Cybernetics*, 19(6):1564–1575, 1989. Also available as Technical Report CMU–CS–88–200, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

[29] B.E. John and D. E. Kieras. The GOMS family of analysis techniques: Tools for design and evaluation. Technical Report CMU-CS-94-181, Computer Science Department, Carnegie Mellon University, August 1994.

[30] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[31] F. W. Leberl. Practical issues in softcopy photogrammetric systems. In *Mapping and Remote Sensing Tools for the 21st Century*, pages 223–230, Washington, D. C., August 1994. ASPRS.

[32] C. Lin, A. Huertas, and R. Nevatia. Detection of buildings using perceptual grouping and shadows. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 62–69, Seattle, Washington, June 19–23 1994.

[33] Y.-T. Liow and T. Pavlidis. Use of shadows for extracting buildings in aerial images. *Computer Vision, Graphics, and Image Processing*, 49(2):242–277, February 1990.

[34] I. S. MacKenzie and C. Ware. Lag as a determinant of human performance in interactive systems. In *INTERCHI'93: Conference Proceedings*, pages 488–493, Amsterdam, April 1993. ACM/SIGCHI, ACM Press.

[35] Chris McGlone. Bundle adjustment with object space constraints for site modeling. In *Proceedings of the SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, volume 2486, pages 25–36, April 1995.

[36] J. C. McGlone and J. A. Shufelt. Projective and object space geometry for monocular building extraction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 54–61, Seattle, Washington, June 19–23 1994.

[37] J. C. McGlone and J. A. Shufelt. Projective and object space geometry for monocular building extraction. Technical Report CMU–CS–94–118, Computer Science Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, 1994.

[38] D. McKeown and Y. Hsieh. Hierarchical waveform matching: A new feature-based stereo technique. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 513–519, Urbana Champaign, Illinois, June 16-18 1992.

[39] D. McKeown and J. C. McGlone. Integration of photogrammetric cues into cartographic feature extraction. In *Proceedings of the SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision*, volume 1944, pages 2–15, September 1993.

[40] David M. McKeown, Jr. Top ten lessons learned in automated cartography. In *International Archives of Photogrammetry and Remote Sensing: Spatial Information from Digital Photogrammetry and Computer Vision*, volume 30, 3/1, Munich, Germany, September 5–9 1994. Also appears in theProceedings of the ARPA Image Understanding Workshop, February 1996.

[41] Edward M. Mikhail. *Observations and Least Squares.* Harper and Row, New York, 1980.

[42] R. Mohan and R. Nevatia. Using perceptual organization to extract 3-D structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11), November 1989.

[43] W. J. Mueller and J. A. Olson. Model-based feature extraction. In E. B. Barrett and D. M McKeown, editors, *Integrating Photogrammetric Techniques with Scene Analysus and Machine Vision*, volume 1944, pages 263–272, Orlando, Florida, April 1993. SPIE.

[44] J. L. Mundy, T. Binford, T. Boult, A. Hanson, R. Beveridge, R. Haralick, V. Ramesh, C. Kohl, D. Lawton, D. Morgan, K. Price, and T. Strat. The RADIUS common development environment. In *Proceedings of the ARPA Image Understanding Workshop*, pages 215–226, San Mateo, CA., January 1992. DARPA, Morgan Kaufmann.

[45] A. Newell and H. A. Simon. *Human Problem Solving.* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1972.

[46] D. W. Paglieroni, G. E. Ford, and E. M. Tsujimoto. The position-orientation masking approach to parametric search for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):740–747, July 1994.

[47] D. K. Pathak. Agent design for automatic use of a software system: A case study with a SOAR agent for Mathematica. Technical Report CMU-RI-TR-93-30, The Robotics Institute, Carnegie Mellon University, May 1993.

[48] L. H. Quam and T. M. Strat. SRI image understanding research in cartographic feature extraction. In H. Ebner, D. Fritsch, and C. Heipke, editors, *Digital Photogrammetric Systems*, pages 111–122, Karlsruhe, Germany, 1991. Wichmann.

[49] J. R. Rhyne and C. G. Wolf. Recognition-based user interfaces. In *Advances in Human Computer Interaction*, volume 4, pages 191–250. Ablex Publishing Corporation, Norwood, New Jersey, 1993.

[50] M. Roux, Y. C. Hsieh, and D. M. McKeown, Jr. Performance analysis of object space matching for building extraction using several images. In *Proceedings of the SPIE: Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II*, volume 2486, pages 277–297, 1995.

[51] M. Roux and D. M. McKeown, Jr. Feature matching for building extraction from multiple views. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 46–53, Seattle, Washington, June 19–23 1994.

[52] P. J. Santos, A. J. Baltzer, A. N. Badre, R. L. Henneman, and M. S. Miller. On handwriting recognition system performance: Some experimental results. In *Proceedings of the Human Factors Society 36th Annual Meeting*, pages 283–287, Atlanta, Georgia, October 1992. Human Factors Society.

[53] B. Shackel. Usability - context, framework, definition, design and evaluation. In B. Shackel and S.J. Richardson, editors, *Human Factors for Informatics Usability*, pages 21–37. Cambridge University Press, Cambridge, 1991.

[54] J. A. Shufelt and D. M. McKeown. Fusion of monocular cues to detect man-made structures in aerial imagery. *Computer Vision, Graphics, and Image Processing*, 57(3):307–330, May 1993.

[55] G. B. Tilley. Interactive feature extraction using softcopy photogrammetry. In *Mapping and Remote Sensing Tools for the 21st Century*, pages 188–192, Washington, D. C., August 1994. ASPRS.

[56] J. C. Trinder. The effects of photographic noise on pointing precision, detection, and recognition. *Photogrammetric Engineering and Remote Sensing*, 48(10):1563–1575, October 1982.

[57] J. C. Trinder. Precision of digital target location. *Photogrammetric Engineering and Remote Sensing*, 55(6):883–886, June 1989.

[58] J. E. Unruh and E. M. Mikhail. Mensuration tests using digital images. *Photogrammetric Engineering and Remote Sensing*, 48(8):1343–1349, August 1982.

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890