Groundtruthing The RADIUS Model-Board Imagery

Ken Thornton, Desika C. Nadadur, Visvanathan Ramesh, Xuefi Liu
Xining Zhang, Anand Bedekar, Robert Haralick*

Department of Electrical Engineering
University of Washington
Seattle, Washington 98195
E-mail: {thornton, desika, rameshv, xuefi, xzhang, anand, haralick}@george.ee.washington.edu

Abstract
Our activities over the last year have involved
(1) constructing 3D polyhedral site-models for the
model-boards, (2) obtaining the exterior orientation
parameters for all the model-board imagery, and (3)
groundtruthing features of interest and features not of
interest in the model-board imagery. We have also
begun tuning algorithms for image feature extraction
and classification that use information from distribu-
tions obtained from the ground-truth data.

1 Introduction
The RADIUS project involves developing algo-
rithms for (1) 3D site-model construction, (2) 3D site-
model to image registration, and (3) change detection
and site-model updating. Our activities over the past
year have involved efforts in generating ground-truth
data for the RADIUS model-board image datasets so
that algorithms for each of these tasks may be evalu-
ated. This document describes the protocol we fol-
lowed in preparing the ground-truth data. An an-
nouncement regarding an ftp-site location for retriev-
ing this data will be made at the workshop.

The RADIUS model-board dataset currently con-
ists of 78 images, 38 of model-board 1 and 40 of
model-board 2. The model-boards were constructed
by ITEK Optical Systems and are roughly 40 inches
by 40 inches in size, where 1 inch on the model cor-
responds to 500 inches in the simulated scene. The
images are typically 1038 pixels/row by 1320 pix-
els/column and were acquired by a Kodak “Mega-
pixel” digital camera from a distance of approximately
260 inches. Figure 1 illustrates a typical model-board
2 image. We were also provided files illustrating the
layout of the model-boards. Figure 2 illustrates the
layout of model-board 2.

The model-board 1 dataset consists of 28 “J” im-
age and 10 “K” images. The model-board 2 dataset
consists of 40 “M” images. The resolution of the im-
agery varies from high-resolution (focal length of ap-
proximately 61 mm) to low-resolution (focal length of
27.2 mm). Approximate image-acquisition geometry,

*Funding from DARPA contract 92-F1428000-000 is grate-
fully acknowledged.
including sun and camera angles relative to the model-board, was provided.

2 The Purpose of Image Annotations

Depending on the nature of the RADIUS task, whether it be site-model construction (building extraction), site-model to image registration, or site-model change-detection/image analysis, the nature and goals of the annotation process will vary.

For example, with the RADIUS dataset we were given a number of 3D control points. For the purpose of camera resection, the annotation process involves identifying the corresponding image points.

Our initial site-models consist only of polyhedral buildings. Thus, in the complete image-annotation process described below, the features of interest were those features belonging to buildings.

For evaluating the low-level feature extraction algorithms used by algorithms that automatically extract buildings and register site-models, more elaborate annotation is needed. For example, if the main purpose is to evaluate graytone corner finders, a user should label all possible locations where he/she perceives a gray scale corner. On the other hand, if the main purpose if to evaluate edge finders, the user may identify edge pixels where he/she perceives an intensity edge. One question that arises is: should intensity edges due to shadows be labelled as groundtruth edges? The answer to this question is “yes.” The boundaries detected that are part of shadows are legitimate intensity edges and should be used as groundtruth in an edge finder evaluation. On the other hand, when the focus shifts towards analysis of matching modules which may not use shadow information, one will have to treat the segments that arise from shadows as extraneous information. This information may be used to compute statistics on the clutter around instances of the ideal model.

The main point is that the annotation process will have to specify information required for performing end-to-end evaluation as well as component evaluation. Depending on the nature of the evaluation, the interpretation of the annotated data changes. During some evaluations, parts of the annotated data may be ignored.

3 Activities Related to Site-Model Construction

We were not provided a complete site-model for the model-board imagery. Instead, we were given the coordinates of 118 3D control points for model-board 1 and 138 3D control points for model-board 2. Most, but not all, of these points belong to buildings.

Each model-board contains approximately 50 buildings. Our initial activities have involved estimating the exterior orientation for each of the model-board images and then creating triangulated 3D models of these buildings for the model-board 1 site. In doing so we have (1) itemized a list of approximately 400 additional points for which we need 3D coordinates (building corner points), (2) determined the image positions, if possible, of these points in each of the 10 "K" images, and (3) generated by triangulation a least-squares 3D position for each point. This gives an initial location for each of the additional 400 control points that can then be fine-tuned using the multi-image resection algorithm described in Section 4.2 below.

Our initial site-model consists of polyhedral buildings. The RADIUS community is also developing algorithms for the automatic construction of buildings from multi-view imagery. These models will have positional and topological errors, e.g. buildings are located inaccurately, are missed completely, added, merged, etc. The nature of evaluating these automatically constructed models is discussed below.

3.1 Estimating 3D Control Points

Once we have identified the image points corresponding to the control point whose position is unknown, we have a bundle of rays that should intersect in a single point (the position of the control point).

We determine by least-squares the 3D point that minimises the sum-of-squared distances to the rays. (The multi-image resection algorithm described in Section 4.2 will compute this approximate 3D point by minimising the sum of the projected errors on the image).

A 3D point $p_n$ on the $n^{th}$ ray has coordinates $p_n = a_n + \lambda_n b_n$, where $a_n$ is the closest point on the ray from the origin, $b_n$ is the direction of the ray (a unit vector), and $\lambda_n$ is a scalar distance along the ray. The vectors $a_n$ and $b_n$ are orthogonal. Given $n$ rays, it can be shown (see the appendix) that the point $p$ that minimises the sum-of-squared distances from the rays can be determined by solving the system

$$\sum_n (I - b_n b_n^T) p = \sum_n a_n.$$

For each of the 400 new control points, we have up to $n = 10$ rays from the 10 "K" images. From these sets of control points we have constructed triangulated 3D models of the buildings in model-board 1.

3.2 Evaluating Site Models

When an algorithm for generating buildings from aerial images is given, we would like to know how the algorithm works on a given data set. The performance is measured in terms of probabilities of correct detection, probability of misdetection, and false alarm ratio.

An aerial image of an urban area usually contains a group of buildings, some roads and trees. Our site models currently consist of polyhedral models of the buildings. Each building in the site model is defined at four levels: polyhedron, surface, line and point. At the polyhedron level, a building has two associated attributes: the building center and building volume. Each building consists of a set of surfaces. Surfaces have three types of attributes: the center, size, and normal pointing away from the building. Each surface is bounded by a set of line segments. Each line segment has three attributes: center, length and direction cosine. The end points of the line segments consist of the point level specification.
To evaluate the performance of a building extraction algorithm, the ground truth site model is compared against the extracted site model. To do this, we establish a metric to compare 2 such buildings, one from the groundtruth site and one from the extracted site. Assuming bounding constraints on this metric, we then search for the best correspondence between extracted buildings and groundtruth buildings. From this correspondence, we may calculate the probability of correct detection, the probability of misdetection, and the probability of a target having no corresponding groundtruth target. Preliminary details may be found in (Haralick, Ramesh, and Liu, 1994).

4 Activities Related to Site-Model to Image Registration

We have determined by hand the correspondences between the 118 known 3D control points and image points for each of the 38 model-board 1 images and between the 138 known 3D control points and image points for each of the 40 model-board 2 images. Not all of the control points were visible in each image. Because it was likely that human errors occurred in the matching, we ran a robust exterior orientation algorithm to estimate the camera position and orientation. The tables in the appendix give the June 1994 results. We are currently running a full camera calibration on all the images (this includes the 6 exterior orientation parameters plus 5 interior orientation parameters) and expect to have updated results at the workshop. We will also have results comparing our technique to a publicly available implementation of the “Tsai” algorithm (Tsai, 1987; Lens and Tsai, 1988).

4.1 Calibration of Camera

The mathematical model for camera calibration is a combination of the models for interior orientation and exterior orientation. The model for exterior orientation is given by

\[
\begin{pmatrix}
u_a \\
v_p
\end{pmatrix} = f \begin{pmatrix}
p \\
q
\end{pmatrix}
\]

\[
\begin{pmatrix}
p \\
q
\end{pmatrix} = R(a, b, c) \begin{pmatrix}
x - z_0 \\
y - y_0 \\
z - z_0
\end{pmatrix}
\]

where \((u_a, v_p)\) is the sensor plane position of the projection of an object point and \((p, q, s)\) and \((z, y, z)\) are the position of the object point in camera and object coordinates, respectively. The unknown parameters consist of the position \((x_0, y_0, z_0)\) of the camera in object coordinates and the rotation parameters \(a, b, c\) necessary to align the object coordinate system with the camera coordinate system. We parameterize the rotation matrix using quaternions (Haralick and Shapiro, Volume 2), and then convert to the standard photogrammetric system described in (Mikhail, 1994).

Usually, \(p, q, s, z, y, z, x_0, y_0, z_0\) are measured in units of cm or m and \(f, u_a, v_p\) are given in units of mm.

The interior orientation of a camera specifies those parameters needed to recover the sensor plane coordinates \((u_p, v_p)\) of a point from the measured frame-buffer coordinates \((u_f, v_f)\), plus the camera constant \(f\). The frame-buffer coordinates are measured in units of pixels, which we will abbreviate by px. Three steps take us from \((u_p, v_p)\) to \((u_f, v_f)\): radial lens distortion, axis scaling, and principal point shift. We discuss each step in turn.

4.1.1 Radial Lens Distortion

Consider a radially symmetric lens distortion acting on the point \((u_p, v_p)\) to produce the point \((u_d, v_d)\). Let \((\Delta u_d, \Delta v_d) = (u_d - u_p, v_d - v_p)\) be the vector of corrections which when added to \((u_d, v_d)\) give \((u_p, v_p)\).

If \(r_d\) and \(r_p\) are the distances of \((u_d, v_d)\) and \((u_p, v_p)\) from the origin of the sensor coordinate system, we have \(r_d^2 = u_d^2 + v_d^2\) and \(r_p^2 = u_p^2 + v_p^2\). If \(\Delta u_d\) and \(\Delta v_d\) are small, we can make the following approximations

\[
\begin{align*}
r_d^2 &= u_d^2 + v_d^2 \\
(r_d + \Delta r_d)^2 &= (u_d + \Delta u_d)^2 + (v_d + \Delta v_d)^2 \\
r_d^2 &+ 2r_d\Delta r_d \\ &= u_d^2 + 2u_d\Delta u_d + v_d^2 + 2v_d\Delta v_d \\
\Delta r_d &= u_d\Delta u_d + v_d\Delta v_d
\end{align*}
\]

By similar triangles, the relation \(\Delta u_d u_d = \Delta v_d v_d\) must hold exactly. Substituting for \(\Delta u_d\) and \(\Delta v_d\) above, we have that

\[
\begin{align*}
r_d\Delta r_d &= u_d\left(\frac{\Delta u_d u_d}{u_d}\right) + v_d\left(\frac{\Delta v_d v_d}{u_d}\right) \\
\frac{\Delta r_d}{r_d} &= \frac{\Delta u_d}{u_d} + \frac{\Delta v_d}{v_d} \\
\frac{\Delta r_d}{r_d} &= \frac{\Delta u_d}{u_d} + \frac{\Delta u_d}{u_d}
\end{align*}
\]

Consider the \(v_d = 0\) axis. In our distortion model, when \(\Delta u_d\) is positive the undistorted point lies further from the origin than the distorted point. Since we are using a radially symmetric model, this requires that if \(\Delta u_d\) is positive at the point \((u_d, 0)\) where \(u_d > 0\), \(\Delta u_d\) must be negative at the point \((-u_d, 0)\) but have the same magnitude. A suitable model for this situation, assuming that \(\Delta u_d = 0\) at the point \((0, 0)\), is

\[
\Delta u_d = k_1 u_d + k_2 u_d^3 + k_3 u_d^5
\]

An entirely analogous argument can be presented for the \(u_d = 0\) axis. Again, the same model holds for \(u_d\). The coefficients must be the same to guarantee symmetry. Thus, we have

\[
\Delta v_d = k_1 v_d + k_2 v_d^3 + k_3 v_d^5
\]

Finally, we can combine these two models into a model which also includes points off the \(u_d = 0\) and \(v_d = 0\) axes by letting

\[
\Delta r_d = k_1 r_d + k_2 r_d^3 + k_3 r_d^5
\]
Using the above results, we have that
\[
\begin{pmatrix}
  u_p \\
  v_p
\end{pmatrix} = \begin{pmatrix}
  u_d + \Delta u_d \\
  v_d + \Delta v_d
\end{pmatrix} = \begin{pmatrix}
  1 + \frac{\Delta r_d}{r_d} \\
  \frac{1}{r_d}
\end{pmatrix} \begin{pmatrix}
  u_d \\
  v_d
\end{pmatrix} = (1 + k_1 + k_2 r_d^2 + k_3 r_d^4) \begin{pmatrix}
  u_d \\
  v_d
\end{pmatrix}.
\]

4.1.2 Axis Scaling

The points \((u_d, v_d)\) and \((u_p, v_p)\) are measured in sensor plane coordinates, usually in units of mm. Now, consider the point \((u_f, v_f)\) in frame-buffer coordinates. We must have
\[
\begin{pmatrix}
  u_d \\
  v_d
\end{pmatrix} = \begin{pmatrix}
  s_u \\
  s_v
\end{pmatrix} \begin{pmatrix}
  u_f \\
  v_f
\end{pmatrix} (8)
\]
where \(s_u\) and \(s_v\) are measured in units of mm/px and \(u_f\) and \(v_f\) are measured in units of px.

4.1.3 Principal Point Shift

The point where the optical axis of the camera pierces the image plane does not always resolve to the point with frame-buffer coordinates \((u_f, v_f) = (0, 0)\)\(^1\). In this case, the relationship between the distorted sensor plane position of the image points and their frame-buffer coordinates will be given by
\[
\begin{pmatrix}
  u_d \\
  v_d
\end{pmatrix} = \begin{pmatrix}
  s_u \\
  s_v
\end{pmatrix} \begin{pmatrix}
  u_f - u_0 \\
  v_f - v_0
\end{pmatrix} (9)
\]
where \((u_0, v_0)\) are the coordinates in pixels of this point of intersection.

4.1.4 Combined Model

For notational convenience, we drop the subscripts and let \((u, v) = (u_f, v_f)\) be the measured frame-buffer coordinates of a perspective projection image point. Our combined interior orientation and exterior orientation model is now
\[
\begin{pmatrix}
  s_u \\
  s_v
\end{pmatrix} \begin{pmatrix}
  u - u_0 \\
  v - v_0
\end{pmatrix} = \begin{pmatrix}
  p \\
  q
\end{pmatrix} = R(a, b, c) \begin{pmatrix}
  z - x_0 \\
  y - y_0 \\
  z - z_0
\end{pmatrix},
\]
where
\[
r^2 = [s_u(u - u_0)]^2 + [s_v(v - v_0)]^2.
\]
Given a set of corresponding 3D object points and 2D frame-buffer points, it is not possible to determine a unique set of parameters \((s_u, s_v, u_0, v_0, k_1, k_2, k_3, x_0, y_0, z_0, a, b, c, f)\). First, notice that we can multiply \(f\) by any non-zero constant, and divide \(s_u\) and \(s_v\) by the same constant without modifying the model. For this reason, we assume that \(s_v\), the vertical scaling from px to mm is known. This constant can usually be obtained from the data sheets of the sensor manufacturer. The number of unknown parameters is now 13 instead of 14. However, even if \(s_v\) is assumed known, notice that the model could be written as
\[
\begin{pmatrix}
  p_4 \\
  p_5 \\
  p_6
\end{pmatrix} = R(a, b, c) \begin{pmatrix}
  p \\
  q \\
  s
\end{pmatrix} \begin{pmatrix}
  u - u_0 \\
  v - v_0
\end{pmatrix} = \begin{pmatrix}
  p_4 \\
  p_5 \\
  p_6
\end{pmatrix} = R(a, b, c) \begin{pmatrix}
  z - x_0 \\
  y - y_0 \\
  z - z_0
\end{pmatrix},
\]
where \(p_4 = \frac{1 + k_1}{r_1} \), \(p_5 = \frac{k_2}{r_2} \), \(p_6 = \frac{k_3}{r_3} \), and \(p_4 = \frac{ru}{w} \).
Here, we see that the number of unknown parameters is 12. To remedy this situation, we will increase the order of the radial distortion from that specified above to
\[
\Delta r_d = k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6.
\]
The complete model consisting of the parameters \((s_u, u_0, v_0, k_1, k_2, k_3, x_0, y_0, z_0, a, b, c, f)\) may now be written as
\[
\begin{pmatrix}
  p \\
  q \\
  s
\end{pmatrix} = R(a, b, c) \begin{pmatrix}
  z - x_0 \\
  y - y_0 \\
  z - z_0
\end{pmatrix}.
\]
For convenience, we will write the model as
\[
g(x, \theta_1) = h(X, \theta_2)
\]
where
\[
\begin{align*}
x &= (u_f, v_f)' \\
X &= (x, y, z)' \\
\theta_1 &= (s_u, u_0, v_0, k_1, k_2, k_3)' \\
\theta_2 &= (x_0, y_0, z_0, a, b, c, f)'.
\end{align*}
\]
For convenience, we have grouped the focal length with the exterior orientation parameters. In practice, as in (Tsai, 1987), we use only 1 radial distortion parameter, reducing the total number of unknowns to 11.

4.1.5 Linearization

In estimating the vectors of parameters \(\theta_1\) and \(\theta_2\) given initial estimates and noisy observed values of the 2D
frame-buffer coordinates, we linearize the model and iteratively solve for the corrections to the model parameters. It is simplest to find the parameters $\theta_1$ and $\theta_2$ to minimize

$$\sum_n \|g(x_n, \theta_1) - h(x_n, \theta_2)\|^2.$$ 

In this case, we linearize each side of the equation with respect to some initial parameters and find at each iteration the corrections $\Delta \theta_1$ and $\Delta \theta_2$ to minimize

$$\sum_n \|(A_n + B_n \Delta \theta_1) - (C_n + D_n \Delta \theta_2)\|^2.$$ 

4.1.6 The Robust Implementation

If some of the corresponding point pairs are incorrect, the standard equally weighted least squares (EWLS) technique is known to produce poor results. In our robust technique, at each iteration, we classify all the points as inliers or outliers by looking at the robust Mahalanobis distances of each point.

Let $r = (r_1, \ldots, r_n)$ be the set of 2D residuals in sensor coordinates. In order to classify point $n$, we first determine its robust Mahalanobis distance. Given a set $\{r_1, \ldots, r_n\}$ of points with sample average $\bar{r}$ and sample covariance $S$ (assumed to be positive-definite), the sample Mahalanobis distance of any point $n$ from the average $\bar{r}$ is defined as

$$m_n = \left( (r_n - \bar{r})' S^{-1} (r_n - \bar{r}) \right)^{\frac{1}{2}}.$$

We have used the projection algorithm described in Rousseeuw and van Zomeren (1990) to obtain robust Mahalanobis distances. It is also possible, of course, to estimate these distances directly if robust estimates of multivariate mean and covariance are available.

Let $M = (m_1^2, \ldots, m_n^2)$ be the set of all univariate distances. The inliers from this set should be distributed as $\chi^2$. If a random variable $i$ is distributed as $\chi^2$, then we know that

$$\Pr(i \leq 9.2103) = 0.990,$$
$$\Pr(i \leq 7.3778) = 0.975,$$
$$\Pr(i \leq 5.9915) = 0.950,$$
$$\Pr(i \leq 4.6052) = 0.900.$$

In our latest experiments, we have used a rejection threshold of 4.6052.

The robust algorithm is then

(1) Get initial guess $\Theta$.

(2) Form residuals $r_n$ and determine the average sum-of-squared errors $\epsilon^2$.

(3) Determine univariate robust squared Mahalanobis distances $m_n^2$.

(4) Classify points as inliers and outliers.

(5) Determine $\Delta \Theta$, the correction to the current parameter vector, using equally-weighted least-squares with the inliers.

(6) Determine new residuals $r_n$ and new $\epsilon^2$.

(7) If a fixed number of iterations is reached or if the new $\epsilon^2$ is greater than the old $\epsilon^2$, stop. Otherwise, go to (3).

4.2 Multi-image Constrained Calibration

Passpoints are points located in multiple images whose 3D coordinates are only approximately known. For the 10 "K" images, as described above, we have roughly 400 passpoints. We would like to perform a multi-image calibration to estimate adjustments to the coordinates of each passpoint (1200 unknowns), plus the exterior and interior orientation of each camera. We can assume that the principal point $(u_0, v_0)$, image scaling $s_u$, and radial distortion $k_1$ is the same for all the images. We also know that certain groups of images should have the same camera constant $f$. For the 10 "K" images, this amounts to a least squares calculation involving 1287 unknowns (1200 due to passpoints, 60 due to exterior orientation for the 10 images, 3 focal lengths, and 4 interior orientation parameters assuming only 1 radial distortion parameter) subject to linear constraints. We hope to integrate this multiple image calibration software into RCDE.

4.3 Estimation of Sun Position

For the model-board imagery, the sun is a finite distance away. For those 3D control points that cast strong shadows, if we identify corresponding shadow points on the ground, we will have a bundle of rays that should intersect at the sun. Using the same least-squares calculation described above for estimating control points from multiple images, we can estimate the 3D location of the sun.

5 Activities Related to Change Detection/Image Analysis

We have modified a public-domain tool "Xfig" to allow us to label features of interest and non-intersect in the RADIUS imagery. By measuring the distributions of these features, we have developed a methodology for setting the parameters of image-processing algorithms. That methodology is the subject of another paper at this workshop (Ramesh, 1994). This section describes the tool, the protocol we have followed in performing the labelling, and the distributions that we measure for the purpose of setting the parameters of edge and line finding algorithms.

5.1 Groundtruth Generation from Model Board Imagery

The following subsections describe the manual generation of groundtruth labelled boundaries using the Xfig graphics interface. The public domain graphics tool Xfig, which supported only fig format, was modified to handle gray scale images for the purpose of generating the groundtruth. Currently, Xfig includes text primitives and geometric primitives such as ellipses, circles, polygons, polylines, and open and
closed splines. It can group primitives to form a compound object and break a compound object into its primitives. It also includes features to move, copy, rotate, scale, and zoom objects.

5.1.1 Features in the imagery

Although there are many "structures" present in the model-board imagery, e.g., buildings, roads, railways, trees, we have initially constructed site models consisting solely of buildings. Thus, in our annotation protocol, only buildings are features of interest. Buildings appear in various ways in the imagery. We have instructed the annotators to outline buildings in order of increasing visual complexity. Examples of these different complexity classes include the following.

1. An isolated building with a simple flat roof. An isolated building does not occlude and is not occluded by any other object in the 2D image. An isolated building is shown in Figure 3.

2. A non-isolated buildings with a simple flat roof. Buildings with simple flat roofs that come into contact with another building in the 2D image are called non-isolated buildings with simple flat roofs.

3. Buildings with complex roofs. A building with complex roof structures is referred to as a building with complex roof. An example is given in Figure 4.

Figure 5 illustrates non-isolated buildings with simple and complex roofs. Figure 6 gives an illustration of features of non-interest. We also refer to the model-board layout (see Figure 2) and other images when outlining buildings upon which shadows are cast by other buildings.

5.1.2 Annotation protocol

The purpose of the annotation protocol is to establish uniform procedures by which the same annotation will be produced regardless of who the person doing it may be.

The process of annotating features of interest can be divided into outlining isolated buildings with simple flat roofs, outlining non-isolated buildings, and outlining buildings with complex roofs. Outlining the isolated buildings with simple flat roofs proceeds from outlining the roof to outlining the walls, in that order. Figure 7 illustrates the groundtruth for an isolated building with a simple flat roof. If the extent of occlusion in non-isolated buildings is ordered from least-to-most occlusion, drawing should proceed from the least-to-most occluded building. For example, see Figure 8. In annotating buildings with complex roof structures, if the roof structures do not occlude the main roof, then annotation begins at a main roof boundary; otherwise, it begins at the structures on the roof taking into account their occlusions with one another, using the procedure discussed for the other two types of buildings. Figure 9 illustrates the annotation for a building with complex roof.

Features of non-interest consist of parks, roads, railway tracks, shadow regions, etc. Figure 10 shows the annotation for features of non-interest.
In preparing the database, outlines for each building are logically grouped together and given a label corresponding to that in the model board layout. Features of non-interest are not grouped together and are not given a label.

Figure 11 illustrates the annotation for a model-board image. Figure 12 illustrates the annotation overlayed on the model-board image. See Nadadur and Zhang (1994) for the complete annotation protocol.

5.2 Distributions from groundtruth annotations

In this section we describe the nature of the groundtruth data employed during the evaluation of edge and line finders. In addition, we provide details of the statistics that are computed using the groundtruth.

The edge and line finding schemes that we are currently evaluating are based on intensity gradients. Therefore, for each image in the dataset, we ask a user to outline all pixels that he/she perceives to be an intensity edge. Clearly, the groundtruth that is generated will depend largely on the user performing the outlining. We ask the user to outline all intensity edges that he/she could see in the displayed image\(^2\). Below, we will outline details of the statistics that we wish to compute. Specific details of how these statistics are used to select algorithm parameters are discussed in another paper (Ramesh, 1994). Distributions that may be computed from the groundtruth edge image and the original gray scale image, include:

1. Gray level noise variance. One of the assumptions that is often made by feature extraction operators is that the ideal data in each local neighborhood of the operator is corrupted with additive Gaussian noise. We wish to determine the distribution of the gray level noise variance parameter \(\sigma^2\) from the annotated data.

2. Edge Gradient distribution. Due to illumination effects, edges with various contrast are present in an image. The edge gradient distribution gives us a measure of the variation of the intensity gradients in the population of images. Associated with an ideal edge is a scale (for example, the width of the ramp) and the contrast. Ideally, we would like to compute, from the groundtruth, the edge scale distribution as well as the associated contrast. The estimate for the gradient value is then obtained at the appropriate scale. The distribution of edge gradients and edge scales will be utilized in the selection of the appropriate edge operator neighborhood size and gradient threshold. In addition to the edge scale distribution, the nearest neighbor distance distribution between edges should be obtained from the groundtruth data. We define the term “nearest neighbor edge pixel”

\(^2\)One way of removing the subjectivity in the outlining process is by computing an average groundtruth edge image from the labellings obtained from multiple users. See (Ramesh, 1994).
for a groundtruth edge pixel as the closest edge pixel along the direction of the edge intensity profile. The distance distribution between nearest neighbors specify the separation between adjacent edges. This distribution can be computed directly from the groundtruth data.

3. Edge direction distribution. In certain applications, the imaging may be controlled, or perhaps not all viewing configurations are allowed. In these situations, the distribution of the imaging parameters induce a prior distribution for the geometric parameters of the image features. For example, for features of interest, not all edge orientations may be likely. We wish to incorporate domain specific information, such as the prior distribution of the edge orientation, in the feature extraction algorithm.

4. Edge Profile. Associated with an intensity edge is the function that models the intensity profile. The edge profile information provides insight into what the appropriate edge model is and what algorithm is appropriate for precise localization of the edge.

5. Gradient variation along boundary. Although the statistics above provide an adequate description of the characteristics of edges, we wish to go one level further. In order to cope with graytone variations due to illumination, and reliably detect edges in areas of low contrast, some adaptive edge gradient thresholding scheme may be necessary. Previously, a renewal process model (Ramesh, 1992) was developed to describe the breakage of a boundary into pieces. The distributions of the edge segments and the gaps were modelled as exponential distributions. These distributions are exponential only under the assumption of independence between neighboring pixel edge gradient estimates and that the true edge gradient was a constant along the boundary. In reality, the gradient along the boundary varies in a smooth fashion. We wish to estimate the edge gradient profile along object boundaries with the idea of utilizing the profile to derive more appropriate models for the segment and gap length distributions. In fact, one can utilize the edge profile information to perform edge linking.

We have described the distributions of parameters relevant to edgel extraction. Prior distributions for parameters describing the geometry of the 2D contours in the image population may also be derived from the annotations. For example, one can derive priors for the joint distribution of parameters of lines in polygonal approximations to the 2D contours. Once the groundtruth 3D site model is available, we could derive appropriate probability distributions for the parameters describing the class of buildings: the building heights, widths, lengths and spatial position. We can think of deriving a population of images of the given site by capturing images at various positions and orientations of an imaging sensor. One could add time as another parameter as it gives an idea of where the light source (in this case the sun) is with respect to the world. Thus, the imaging sensor has associated with it the free variables of position and viewing light. The imaging sensor has intrinsic parameters that we assume are not variable. For example, in the case of a simple pin-hole camera model the intrinsic parameters include the focal length, dimensions of the CCD array, and the number of rows and columns in the sensor array. The distribution of the parameters describing the exterior orientation of the imaging sensor can be derived from the annotations as well.

All of the distributions in this section describe the nature of the image population. By combining these prior distributions with a complete performance analysis of each component to derive performance measures of the system as a function of the tuning constants of the system. We are geared towards choosing the tuning constants to optimize the performance of the total system.

6 Conclusion
We have described the methodology we have followed in groundtruthing the RADIUS imagery. The groundtruthing has involved (1) matching control
points to image points for the purpose of exterior orientation, (2) matching image points to image points for the purpose of site-model construction, and (3) delineating features of interest and non-interest for the purpose of measuring appropriate distributions and deriving from those distributions the parameters of various image-processing algorithms.

As described in (Hoogs and Hackett, 1994), there are additional variables related to image context, geometric context, temporal context, radiometric context, and functionality context that influence the selection of appropriate image understanding algorithms and the setting of parameters of those algorithms. The site-model incorporates most of the geometric context. Future work will involve modelling temporal and radiometric contexts.

### Appendix 1: Least-Squares Ray Intersection

Obtaining initial estimates for the 3D positions of control points by triangulation and obtaining estimates for the 3D sun location requiring solving the problem of determining the point \( p \) that minimizes the sum-of-squared distances to a set of rays. In this section we give a short derivation of the result presented in the paper.

Let \( p_n \) be a point on the \( n \)th ray. It has coordinates \( p_n = a_n + \lambda_n b_n \), where \( a_n \) is the closest point on the ray from the origin, and \( b_n \) is the vector representing the direction cosines of the ray (a unit vector). Let \( p \) be a point not on the ray. Then \( ||(I - b_n b_n')(p - a_n)|| \) represents the length of the component of \( p - a_n \) in the direction orthogonal to \( b_n \). The error measure to be minimized is then

\[
\epsilon^2 = \sum_n ||(I - b_n b_n')(p - a_n)||^2
\]

\[
= \sum_n (p - a_n)' (I - b_n b_n') (p - a_n).
\]

Then \( \frac{\partial \epsilon^2}{\partial p} = 0 \) implies

\[
\sum_n 2(I - b_n b_n')(p - a_n) = 0
\]

\[
[\sum_n (I - b_n b_n')] p = \sum_n (I - b_n b_n') a_n
\]

\[
[\sum_n (I - b_n b_n')] p = \sum_n a_n.
\]

### Appendix 2: Resection Results

For each image, the robust procedure rejected certain points. These may have been mismatched or poorly located. The rejected points and the estimated variance of the noise on the inliers is reported below for each image. Updated results should be available at the workshop.

<table>
<thead>
<tr>
<th>Image</th>
<th>Number of Points</th>
<th>Rejected Points</th>
<th>Sigma in Pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>75</td>
<td>8, 15, 80, 82</td>
<td>2.749</td>
</tr>
<tr>
<td>J2</td>
<td>79</td>
<td>66, 80, 90</td>
<td>2.778</td>
</tr>
<tr>
<td>J3</td>
<td>71</td>
<td>8, 15, 19, 80</td>
<td>2.798</td>
</tr>
<tr>
<td>J4</td>
<td>58</td>
<td>66, 80, 90</td>
<td>2.857</td>
</tr>
<tr>
<td>J5</td>
<td>77</td>
<td>80, 90</td>
<td>3.260</td>
</tr>
<tr>
<td>J6</td>
<td>68</td>
<td>6, 66, 80, 90</td>
<td>2.958</td>
</tr>
<tr>
<td>J7</td>
<td>79</td>
<td>66, 80, 90</td>
<td>3.140</td>
</tr>
<tr>
<td>J8</td>
<td>89</td>
<td>3, 4, 5, 5, 7, 12, 66, 80, 90, 100, 116, 119, 132</td>
<td>2.958</td>
</tr>
<tr>
<td>J9</td>
<td>76</td>
<td>6, 27, 66, 80</td>
<td>3.500</td>
</tr>
<tr>
<td>J10</td>
<td>79</td>
<td>7, 82, 66, 75, 80</td>
<td>2.864</td>
</tr>
<tr>
<td>J13</td>
<td>100</td>
<td>7, 14, 24, 57, 60, 66, 80, 90</td>
<td>3.117</td>
</tr>
<tr>
<td>J15</td>
<td>104</td>
<td>6, 66, 80, 90</td>
<td>1.834</td>
</tr>
<tr>
<td>J19</td>
<td>85</td>
<td>6, 7, 14, 15, 37</td>
<td>2.781</td>
</tr>
<tr>
<td>J20</td>
<td>74</td>
<td>18, 80, 90, 121</td>
<td>3.583</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Number of Points</th>
<th>Rejected Points</th>
<th>Sigma in Pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>J21</td>
<td>83</td>
<td>6, 7, 19, 27, 83</td>
<td>2.698</td>
</tr>
<tr>
<td>J23</td>
<td>99</td>
<td>6, 66, 80, 90, 100</td>
<td>2.789</td>
</tr>
<tr>
<td>J24</td>
<td>80</td>
<td>57, 60, 66, 80</td>
<td>2.378</td>
</tr>
<tr>
<td>J25</td>
<td>80</td>
<td>90, 93, 132</td>
<td>2.379</td>
</tr>
<tr>
<td>J26</td>
<td>101</td>
<td>6, 10, 32, 66, 90</td>
<td>2.845</td>
</tr>
<tr>
<td>J27</td>
<td>103</td>
<td>6, 7, 66, 80, 90</td>
<td>2.530</td>
</tr>
<tr>
<td>J28</td>
<td>89</td>
<td>5, 20, 66, 80, 90</td>
<td>2.580</td>
</tr>
<tr>
<td>J29</td>
<td>97</td>
<td>15, 73, 80, 90, 110</td>
<td>2.695</td>
</tr>
<tr>
<td>J30</td>
<td>94</td>
<td>15, 80, 90, 121</td>
<td>2.787</td>
</tr>
<tr>
<td>J31</td>
<td>104</td>
<td>14, 66, 80, 82, 90, 100, 101, 102, 132</td>
<td>2.430</td>
</tr>
<tr>
<td>J32</td>
<td>101</td>
<td>7, 66, 80, 90, 93</td>
<td>2.341</td>
</tr>
<tr>
<td>J33</td>
<td>104</td>
<td>66, 80, 90, 123</td>
<td>1.879</td>
</tr>
<tr>
<td>J34</td>
<td>107</td>
<td>66, 80, 90</td>
<td>1.814</td>
</tr>
<tr>
<td>J35</td>
<td>107</td>
<td>66, 90</td>
<td>2.083</td>
</tr>
<tr>
<td>Image</td>
<td>Number of Points</td>
<td>Rejected Points</td>
<td>Sigma in Pixels</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>K1</td>
<td>35</td>
<td></td>
<td>4.139</td>
</tr>
<tr>
<td>K2</td>
<td>48</td>
<td>15, 80</td>
<td>3.254</td>
</tr>
<tr>
<td>K3</td>
<td>84</td>
<td>1, 71, 80, 90</td>
<td>2.517</td>
</tr>
<tr>
<td>K4</td>
<td>33</td>
<td>90, 95, 100, 129</td>
<td>3.879</td>
</tr>
<tr>
<td>K5</td>
<td>41</td>
<td>36, 43, 45, 80</td>
<td>4.424</td>
</tr>
<tr>
<td>K6</td>
<td>55</td>
<td>66, 80, 90</td>
<td>2.431</td>
</tr>
<tr>
<td>K7</td>
<td>28</td>
<td>80</td>
<td>3.229</td>
</tr>
<tr>
<td>K8</td>
<td>55</td>
<td>15, 75, 80</td>
<td>3.144</td>
</tr>
<tr>
<td>K9</td>
<td>70</td>
<td>15, 48, 80, 90</td>
<td>3.033</td>
</tr>
<tr>
<td>K10</td>
<td>71</td>
<td>6, 48, 75, 80, 90</td>
<td>2.871</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Image</th>
<th>Number of Points</th>
<th>Rejected Points</th>
<th>Sigma in Pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>45</td>
<td>24, 31, 87, 160</td>
<td>1.292</td>
</tr>
<tr>
<td>M2</td>
<td>77</td>
<td>211, 213, 231, 234</td>
<td>2.217</td>
</tr>
<tr>
<td>M3</td>
<td>54</td>
<td>265</td>
<td>2.167</td>
</tr>
<tr>
<td>M4</td>
<td>55</td>
<td>23, 265</td>
<td>2.059</td>
</tr>
<tr>
<td>M5</td>
<td>54</td>
<td>214, 265</td>
<td>2.099</td>
</tr>
<tr>
<td>M6</td>
<td>45</td>
<td>265</td>
<td>1.851</td>
</tr>
<tr>
<td>M7</td>
<td>33</td>
<td>1, 2, 6, 7</td>
<td>2.096</td>
</tr>
<tr>
<td>M8</td>
<td>72</td>
<td>41, 231, 234, 237</td>
<td>2.401</td>
</tr>
<tr>
<td>M9</td>
<td>57</td>
<td>16, 17, 18, 229</td>
<td>2.350</td>
</tr>
<tr>
<td>M10</td>
<td>108</td>
<td>55, 234, 237, 238</td>
<td>2.221</td>
</tr>
<tr>
<td>M11</td>
<td>75</td>
<td>157, 158, 160, 161</td>
<td>1.817</td>
</tr>
<tr>
<td>M12</td>
<td>108</td>
<td>2, 214, 231, 234</td>
<td>2.267</td>
</tr>
<tr>
<td>M13</td>
<td>105</td>
<td>53, 160, 201, 234</td>
<td>1.697</td>
</tr>
<tr>
<td>M14</td>
<td>120</td>
<td>36, 161, 231, 234</td>
<td>1.910</td>
</tr>
<tr>
<td>M15</td>
<td>104</td>
<td>37, 41, 234, 237, 238, 245, 265</td>
<td>1.733</td>
</tr>
<tr>
<td>M16</td>
<td>132</td>
<td>211, 224, 231, 237, 245, 265</td>
<td>1.828</td>
</tr>
<tr>
<td>M17</td>
<td>130</td>
<td>7, 213, 214, 231</td>
<td>1.535</td>
</tr>
<tr>
<td>M18</td>
<td>128</td>
<td>87, 101, 214, 231</td>
<td>1.870</td>
</tr>
<tr>
<td>M19</td>
<td>54</td>
<td>24, 70, 84, 100, 214, 234, 238, 265</td>
<td>2.086</td>
</tr>
<tr>
<td>M20</td>
<td>55</td>
<td>160, 213, 265</td>
<td>2.293</td>
</tr>
</tbody>
</table>

References


