The RADIUS Imagery CDROM Ground Truthed Data Set

Wm. Hunter Hudson
Desika Chari Nadadur
Kenneth B. Thornton
Xufei Liu
Robert M. Haralick
Department of Electrical Engineering
University of Washington
Seattle, WA 98195-2500, U.S.A.
E-mail: {hunter,desika,thornton,haralick}@george.ee.washington.edu
URL: http://george.ee.washington.edu *

Abstract

Keywords: RADIUS, CD-ROM, groundtruth, annotation, corner, map labeling.

This paper describes a CD-ROM containing RADIUS model board imagery and associated groundtruth, for the computer vision community. The RADIUS model board imagery consists of 78 model board aerial images. For each image the boundaries for the features of interest and features of non-interest have been outlined by hand to constitute a groundtruth. This CD-ROM also contains related software and data for the RADIUS imagery. These software projects are a Bayesian corner detector, Bayesian triangulation and point correspondence, model board camera parameters, building parameter estimation, and map labeling. This image set and associated groundtruth will give a common dataset for benchmarking the performance characterization of aerial image understanding.

1 Introduction

Sophisticated computer vision algorithms are composed of many subalgorithms such as corner detectors, edge detectors, segmentation algorithms. Each of these algorithms produce an output that may contain misdetected features of interest and may have falsely detected features. To measure the performance of such feature detector misdetection requires a groundtruth standard. Usual comparisons in the vision community typically compare the output of new algorithms to those of well known algorithms or some visually based procedures where an observer checks the output visually and determines in a qualitative sense if it is satisfactory. Is this the right way to do it? Is it at all correct to assume that the well known algorithms in the field are the Gold-standard? Are the qualitative visually based methods correct?

One solution that seems to be apparent is to obtain a reasonable truth about the world (at least a subset of it) for which these algorithms are being developed. This truth which we call the "Groundtruth" can be used to evaluate the performance of the algorithms that are being developed.

What constitutes such groundtruth? We assume that anything and everything that captures the truth about the world on which an algorithm is to operate constitutes groundtruth. For corner detectors, labeled intersections can be groundtruth. For edge detectors, labeled edges can be groundtruth. To help the computer vision community design better algorithms, the imagery and its groundtruth are being made available on a CD-ROM.

This paper describes the contents of a two volume CD-ROM of RADIUS model board imagery and associated groundtruth for the computer vision community. This CD-ROM has been divided into a two volume set.

Volume I includes 78 RADIUS model board images. For each of these images its associated groundtruth is provided. This groundtruth includes building, shadow, and clutter boundaries delineated and labeled. For each of the two model boards approximately 1000 3D points are identified and located on each of the images in which it appears. The RADIUS provided 3D coordinates are given for approximately 100 control points. Xfig is the annotation tool used for delineating and labeling the images, and is provided.

Volume II includes gradient edge maps for each image with various parameter combinations for the linked edges. A Bayesian corner detector and its detected corner points is included. Multi image Bayesian triangulation software is provided and its associated calculated 3D points are given.

Software for simultaneously refining camera parameter estimates and 3D point locations from multi-image sets of corresponding points is provided. The estimated 3D positions for 876 model board 1 and 532 model board 2 points are given. Interior orientation

*Funding from DARPA contract 92-F1428000-000 is gratefully acknowledged.

camera constants, principal points, and u-scale factors are given. Exterior orientation translation and rotation estimate and their covariance matrices to transform the model board coordinate system to the camera reference frame are given.

The software for building parameter estimation creates 3D points which satisfy the given plane and angle constraints. Both the software and these geometrically corrected building point positions for many building points which satisfy the constraints are provided.

Software to create labeled point and building maps for the model board imagery, and these maps are provided.

2 Volume I - Groundtruth

The procedure for obtaining the ground truth is outlined as follows: Boundary groundtruth constitutes the outlines or boundaries of the various structures found in the image. We use the term "annotation" to refer to this process of outlining the structures. Before one starts annotating an image one should be aware of the contents of the images and what is exact procedure that is to be followed to carry out this process. A systematic procedure is necessary due to the fact that there is an associated variability of appearance of structure from image to image, due to various viewing angles, presence of shadows and such confusing structures.

The following subsections summarize the various structures found in the imagery and the process of annotation. For a detailed discussion of this, an interested reader may refer to [3].

2.1 Classification of structures found in the imagery

We make two main classifications of the structures found in the model board images – *features of interest* and *features of non-interest*. We consider buildings to be of features of interest and other structures such as shadows and other clutter elements as features of non-interest.

The *buildings* (or the features of interest) are further classified as follows:

1. Non-isolated buildings, which are further classified as,
   (a) Buildings that are occluded by other buildings, and
   (b) Buildings that occlude others.
2. Isolated buildings, that are not non-isolated.

The *features of non-interest* are classified as,

1. Clutter (which are the edges of structures such as roads, parks.)
2. Shadows, which are further classified as,
   (a) Shadows due to buildings, and
   (b) Shadows due to clutter.

The following section, briefly discusses the software used for the purpose of annotation.

2.2 Software used for annotation

Xfig public domain software available from MIT has been modified and used to hand draw and label the boundaries. The modified Xfig tool has added capabilities of being able to read in a greyscale image, label the outlines, write a binary image of the boundaries.

The following sections assume that the person annotating the images has knowledge of the use of the Xfig software and that a RADIUS model board image is loaded into the tool and it is ready to be annotated.

2.3 Protocol for annotating the imagery

Annotating the imagery begins with isolated buildings. Each isolated building is annotated by proceeding from roof to walls. The roof boundary is traced first beginning at its top left corner and proceeding clock-wise and then back to the starting point. Then, the walls are outlined from the largest to the smallest proceeding from the top left corner as for the roof. Edges common to structures are traced only once. After the completion of the annotation for each building, all its boundaries are grouped together and given a single unique label.

After completing the annotation for isolated buildings, non-isolated buildings are annotated. A set of non-isolated buildings is annotated proceeding from the least occluded to the most occluded (most visible to the least visible) by following the procedure discussed for annotation of isolated buildings treating each component of the non-isolated building set as if it is isolated. Then each component is labeled as before.

After the completion of the annotations for the buildings, features of non-interest are annotated. Among features of non-interest, (clutter) segments bordering the structures such as parks, and roads are annotated, but no label is given to them. After this, shadows due to the buildings are annotated and the shadow edges corresponding to a particular building are grouped together and given a label that identifies it with that building. After the completion of the annotation of shadow edges due to the buildings, shadow edges due to structures such as parks and roads are annotated and are just labeled as shadow segments not identifying them with any structure.

Care was exercised when annotating structures that fall in the shadows of other structures and become difficult to visualize. Other images that have these structures visible were used as guides in annotating such structures.

After the annotation of a greyscale image is complete, it is stored in two formats, (a) an ASCII format readable by Xfig for future corrections and use by algorithms that work on labeled structures and their coordinate points, (b) a binary image format for algorithms that work on images.

A completed annotation overlayed on a greyscale model board 1 image is shown in Figure 1.

The following subsection describes the components of groundtruth present on the CD-ROM.

2.4 Components of groundtruth on CD-ROM

The CD-ROM contains the following components of the groundtruth.

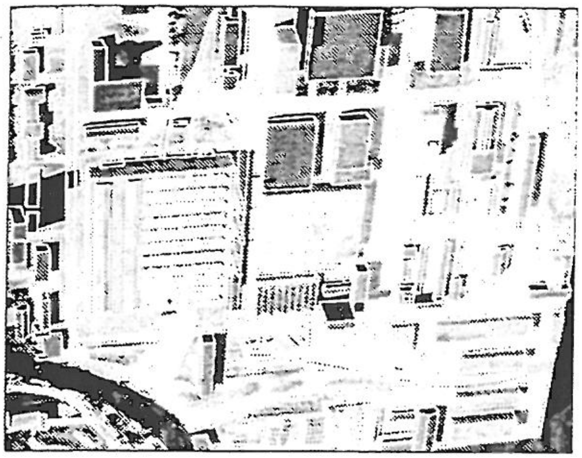

Figure 1: The annotation overlayed on the greyscale image. Annotation is given in *White*.

1. 78 RADIUS model board grey scale images (28 J, 10 K, and 40 M) images.
2. Annotations in ASCII format.
   (a) Annotations for *buildings* only,
   (b) Annotations for *clutter* only,
   (c) Annotations for *shadows* only,
   (d) All three types of annotations *combined*,
3. Annotations in binary image format.
   (a) Annotations for *buildings* only,
   (b) Annotations for *clutter* only,
   (c) Annotations for *shadows* only,
   (d) All three types of annotations *combined*,
4. Shell scripts that separate the *combined* ASCII format annotations into *building*, *clutter* and *shadow* annotations in the same format.
5. Also provided are the printable postscript files of the annotations overlayed on the greyscale images.
6. ASCII files listing the roof and ground points for each building.
7. Corrected RADIUS provided ground control point positions and point IDs.

3 Volume II - Software and Data

The second volume of this CD-ROM set consists of related software and data for the RADIUS model board imagery. The directory structure is divided into five main projects. These projects are:

- A Bayesian Corner Detector
- Bayesian Triangulation and Point Correspondence
- Model Board Camera Parameters
- Building Parameter Estimation
- Map Labeling

These projects are discussed in the following sections.

3.1 A Bayesian Corner Detector

An ideal corner model may be defined by two straight lines and their intersection. When given an observed sequence of ordered points, arising from two line segments, the detected corner is defined as the last observed point arising from the first estimated line segment.

3.1.1 Problem Statement

- Given: an observed sequence of ordered points from an arc segment, $\hat{S} =< \left(\begin{smallmatrix} \hat{r}_i \\ \hat{c}_i \end{smallmatrix} \right) \mid i = 1, ..., I; (\hat{r}_i, \hat{c}_i) \in Z_R \times Z_C >$, where $Z_R \times Z_C$ is the image domain, I is the number of points and $(\hat{r}_i, \hat{c}_i), i = 1, ..., I$ are the results of random perturbations on the points $(r_i, c_i), i = 1, ..., I$ constrained by two lines

$$\begin{aligned} r_i cos\theta_1 + c_i sin\theta_1 - \rho_1 &= 0, \quad i = 1, ..., k; \\ r_i cos\theta_2 + c_i sin\theta_2 - \rho_2 &= 0, \quad i = k+1, ..., I, \end{aligned}$$

where $\theta_j, \rho_j; j = 1, 2$ are line orientation and location parameters for the two line segments. and k is the index of the true corner position (r_k, c_k). Assume perturbations to be independently introduced on each sample point with the Gaussian distributed noise in the direction perpendicular to the line segment. Perturbations on the two line segments can be expressed by

$$\begin{aligned} \hat{r}_i &= r_i + \eta_i cos\theta_1; \\ \hat{c}_i &= c_i + \eta_i sin\theta_1; \quad i = 1, ..., k; \\ \hat{r}_i &= r_i + \eta_i cos\theta_2; \\ \hat{c}_i &= c_i + \eta_i sin\theta_2, ; \quad i = k+1, ..., I. \end{aligned}$$

where $\eta_i \sim N(0, \sigma^2)$.

- Find: the estimated corner $(\hat{r}_{k^*}, \hat{c}_{k^*}), 2 \leq k^* \leq I - 1$, along the arc $\hat{S}$ and the estimates of two line parameters, (θ_1^*, ρ_1^*) and (θ_2^*, ρ_2^*) so that

$$\begin{aligned} &(k^*, \theta_1^*, \rho_1^*, \theta_2^*, \rho_2^*) \\ &= \arg \max_{(k, \theta_1, \rho_1, \theta_2, \rho_2)} P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid \hat{S}, \sigma, I) \end{aligned}$$

By using Bayes theory, the maximization on the posterior probability can be converted to the maximization on the likelihood with prior probability, ie.

$$(k^*, \theta_1^*, \rho_1^*, \theta_2^*, \rho_2^*)$$
$$= \arg \max_{(k,\theta_1,\rho_1,\theta_2,\rho_2)} P(\hat{S} \mid k, \theta_1, \rho_1, \theta_2, \rho_2, \sigma, I)$$
$$\times P(k, \theta_1, \rho_1, \theta_2, \rho_2 \mid \sigma, I)$$

The reader is referred to [9] for a complete description.

Gradient edge map images for the RADIUS J, K, and M images are included in TIFF and ASCII formats.

The subdirectories are divided by the various parameter combination used in generating the images. The nomenclature of the directories is as follows. The first number represents the window side length of the square window used for the Haralick slope-facet operator used for generating gradient information for the edge detector. The second number is the upper threshold for the Canny hysteresis edge linking procedure. The third number is the lower threshold for the Canny edge linker.

Thus in 5-16-2, 5 is the window side length of the square window used for the Haralick slope-facet operator, 16 is the upper threshold for the Canny edge linker, and 2 is the lower threshold.

The output of the corner detector is given in an ASCII format.

3.2 Bayesian Triangulation

The Bayesian triangulation and point correspondence software [8] [7] estimates the 3D coordinates of a point, given its projections in N perspective projection images, and the estimates of the parameters of the N cameras.

3.2.1 Problem Statement

- Given:
  - 2-D points $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N$ are observed in perspective projection images $I_1, \ldots, I_N$ respectively.
  - Estimates $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_N$ of the parameters of the cameras of images $I_1, \ldots, I_N$ respectively are given.
  - The observed points are the result of random perturbations on the perspective projections of a 3-D point $\tilde{q}$ in the respective images, i.e.

    $$\hat{x}_i = P(\tilde{q}, \theta_i) + \xi_i, i = 1, \ldots, N$$

    where $P(\tilde{q}, \hat{\theta}_i)$ denotes the perspective projection of the 3-D point $\tilde{q}$ in a camera with parameter vector θ_i. The random perturbations $\xi_i, i = 1, \ldots, N$ are assumed independent of each other. $\xi_i \sim N(0, \Sigma_{\hat{x}_i}), i = 1, \ldots, N$, where $\Sigma_{\hat{x}_1}, \ldots, \Sigma_{\hat{x}_N}$ are given.
  - The 3-D point $\tilde{q}$ is considered as a random variable with a certain *a priori* density $p(\tilde{q})$.

    $$\hat{\theta}_i = \theta_i + \eta_i, i = 1, \ldots, N$$

    where $\eta_i \sim N(0, \Sigma_{\hat{\theta}_i}), i = 1, \ldots, N$. $\Sigma_{\hat{\theta}_1}, \ldots, \Sigma_{\hat{\theta}_N}$ are given. $\eta_1, \ldots, \eta_N$ are independent of each other.
  - The true camera parameters $\theta_1, \ldots, \theta_N$ are considered as random variables with independent *a priori* densities $p(\theta_1), \ldots, p(\theta_N)$ respectively.
- Find:
  - Estimate $\mathbf{q} = (x, y, z)^T$ to maximize

    $$p(\mathbf{q} \mid \hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N)$$

    i.e. to maximize

    $$p(\mathbf{q}, \hat{x}_1, \ldots, \hat{x}_N, \hat{\theta}_1, \ldots, \hat{\theta}_N)$$

The details of the solution is given in [8] and outlined in [7].

3.3 Model Board Camera Parameters

This section describes the process by which we obtained the camera parameters for the model-board images. Letting (x, y, z) be the world coordinate system, (p, q, s) the camera coordinate system, and (u, v) the image-centered frame-buffer coordinate system, our camera model is

$$\begin{aligned} u &= c * s_u * (p/s) + u_0 \\ v &= c * s_v * (q/s) + v_0, \end{aligned}$$

where

$$\begin{pmatrix} p \\ q \\ s \end{pmatrix} = R(a, b, c, d) \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}.$$

The unknowns are the interior orientation paramters (c, s_u, u_0, v_0) and the exterior orientation parameters $(x_0, y_0, z_0, a, b, c, d)$. Here, c is the camera constant in pixels, s_u is the horizontal image scale factor (s_v is assumed to be 1), (u_0, v_0) is the principal point in pixels, (x_0, y_0, z_0) is the camera location (center of projection), and (a, b, c, d) are the quaternion parameters of the rotation matrix.

Combining the camera parameters into a vector Θ and letting the projection of 3D point x be $T(\mathrm{x}, \Theta)$, the problem of camera calibration, under an i.i.d. Gaussian noise model, is to find Θ to minimize

$$\sum_i ||\mathrm{u_i} - \mathrm{T}(\mathrm{x_i}, \Theta)||^2.$$

This is a nonlinear least squares problem. Given an initial estimate, we iteratively solve for corrections to the camera parameters. A rotation matrix only has 3 degrees of freedom, so we do not solve for 4 quaternion corrections, but for corrections $(\Delta w_1, \Delta w_2, \Delta w_3)$

related to the quaternion by a linear transformation (see [1].)

We obtain an initial estimate for the camera parameters by fixing c to a nominal value, assuming $s_u = 1$ and $u_0 = v_0 = 0$ and solving for the exterior orientation parameters. A robust procedure for doing so is the following. Iterating over all triples of image points whose enclosed triangular area is greater than a threshold, we can determine, in closed form, up to 4 possible solutions for the exterior orientation parameters [2]. For that triple of points, one of the 4 solutions will minimize the median of the distances between the remaining image points and the projections of the model points. Over all the triples, we choose the camera parameters that are associated with the smallest of these medians. The software for choosing an initial guess for the camera parameters, given possibly erroneous matched points, is on the CDROM. We call these camera parameters cp0 to reflect that they are the initial estimates. Previously, we had also adjusted the focal length in obtaining initial estimates. These camera parameters were called cp1, but we found that this step was unnecessary for the model-board data.

Given a set of known 3D model points and these initial guesses cp0, we can then solve individually for the camera parameters for each image. The output is the refined camera parameters, the covariance matrix of the parameters, the covariance matrix of each projected 2D point, and a list of points rejected as outliers. Our procedure for rejecting matches is based on examining robust estimates of the coordinatewise residuals between the observed and projected points. We estimate both the mean and standard deviation of the distribution in order to normalize the residuals. The robust estimate of the mean is the median. The robust estimate of the standard deviation is the related to the median of the absolute deviations about the median. Given a list of numbers $x_1, \ldots, x_n$, we sort then into a new list $x_{(1)}, \ldots, x_{(n)}$. The median, assuming n odd, is $x_{(\frac{n+1}{2})}$. The median of the absolute deviations is the median of the list $\{|x_{(i)} - x_{(\frac{n+1}{2})}|\}$. The robust estimate of the standard deviation is 1.483 times this number. (The constant is chosen so that the estimate is unbiased for Gaussian data.) We then standardize the residuals and reject those above a few standard deviations away from zero. The camera parameters for each individual image, estimated in this manner, are called cp2.

For the model-board data, we had additional points matched between the images for which we did not have corresponding 3D points. Given initial estimates for these 3D points, which we call passpoints, we can solve for their 3D locations by treating both x and Θ as unknowns in the projection model $T(\mathrm{x}, \Theta)$.

Using the camera parameters cp2, we triangulated initial locations based on the following least squares procedure. Let $\mathrm{u_k}$ be a point in image k for which the camera parameters are Θ_k. Associated with that point is the ray $R(\mathrm{u_k}, \Theta_\mathrm{k})$ in world-coordinates. Given a set of matched points between a set of images, it is possible to solve, in closed form, for the 3D point x that minimizes

$$\sum_k d^2(\mathrm{x}, \mathrm{R}(\mathrm{u_k}, \Theta_\mathrm{k})),$$

where $d(\cdot, \cdot)$ is the Euclidean distance between a point and a line. Given the point x we can then project it into each image and measure the error in image coordinates. Because some of the point correspondences may have been invalid, we took all quadruples of images for which we had corresponding points, solved for x, measured the squared errors in the image, and chose x corresponding to the quadruple with the smallest error. Given these initial 3D points, we then refined them using the nonlinear least-squares algorithm discussed above. The initial 3D points given to us we call MB.3d1. The given 3D points plus the initial estimates for the additional points are MB.3d2. The refined estimates are MB.3d3.

Finally, we knew that some of the cameras shared interior orientation parameters. Using the intial camera estimates cp2, we then estimated constrained camera parameters using the initial set of 3D points MB.3d1. We imposed the constraint that the principal points and scale factors were the same among the images, and that certain groups of images had the same focal length. This set of constrained camera parameters was called cp3.

Using the set of camera parameters cp3 and the refined set of 3D points MB.3d3, we then performed a final least-squares estimation where we adjusted both the 3D points and the camera parameters for all the images simultaneously. The final set of estimated 3D points is designated in MB.3d4. The final set of camera parameters is designated in MB.cp4.

3.3.1 Photogrammetry Data on CDROM

Calculated 3D Points

The files contain the following data:

- σ in pixels of the ground control points.
- σ in pixels of the passpoints.
- Point label; chosen consecutively for each point.
- Point flag; for each point, set to a control-point, a passpoint, or not valid.
- The xyz coordinates for each point.
- σ_x^2, σ_y^2, σ_z^2, σ_{xy}, σ_{yz}, and σ_{xz} for each point.

CR Files

These files contain the (column,row) position, σ_c, σ_r, and σ_{cr} for each valid point in each image.

Camera Parameters

For each image, the camera parameters with their covariance matrix is given.

Rejected Points

There is a directory which contains lists of points in each image that are rejected by the calibration procedure.

3pt Files

To get an initial guess for the camera parameters, we started with the point correspondences and then selected triples of image and model points.

Using known interior orientation parameters, we then were able to call a "3-point resection" program to compute initial exterior orientation parameters.

Procedure

In summary, the procedure used to generate the calibration results is

1. Get initial guesses (cp0).
2. Calibrate each image separately using the GCP to get better estimates (cp2).
3. Calibrate all the images together, using constraints on the interior parameters to get new estimates (cp3).
4. Get initial estimates of the passpoints by using the "triangulate" program. This minimizes the sum-of-squared distances in 3D space.
5. Produce better estimates of the passpoints by minimizing the distances in image space.
6. Reestimate, using all the images, both the passpoint locations and the camera parameters. This produces estimates (cp4).

3.4 Building Parameter Estimation

Given the estimated 3D points and their associated covariance matrices, determine the building parameters by fitting the 3D point observations with the building model constraints. We assume that the correspondence between the observations and the points in the building models are established (either from hypothesis or from ground-truth).

3.4.1 Problem Statement

The problem can be described as follows.

- Given
  - partial models of the polygon buildings that includes 3D linear objects (points, lines, planes) and the geometric relations between them.
  - observations of a set of corresponding points. The observations contain the 3D coordinates and the associated covariance matrices.
- Goal :
  - estimate the building parameters (such as point coordinates, plane and line location and orientation) that satisfy the relations in the partial model and are optimal under a given optimality criterion.
  - determine the covariance matrix of the estimated parameters.

To transform this problem into an optimization framework, we need to have the mathematical model that constrains the unknown 3D parameters and the model that links the unknown parameters to the observations. They are the partial mode and the perturbation model.

3.4.2 Partial Building Models

Geometric Relations in Partial Building Models

A site model currently consists of a group of polyhedron building models. Corresponding to the planar surfaces, edges, vertices on a building surface, each building model consists of a set of linear objects (planes, lines and points) and their geometric relations. Sometimes a ground normal vector and its relations with a subset of linear objects are also given.

A partial building model consists of five groups of relations. The first two specify the locations of the linear objects. The other three specify the angles between linear objects and the normal length conditions. Figure 2 and figure 3 illustrates these relations.

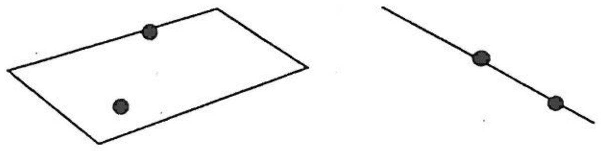

Figure 2: Position relations in a partial model

The first group defines which points are on each plane. Each *point-plane relation* can be described by a planar equation in canonical form. The same point may occur on more than one plane.

The second group specifies which points are on each line, The same point may occur on more than one line.

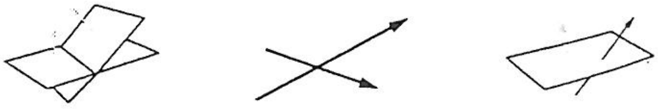

Figure 3: Angle relations in a partial model

The third group, the *plane-plane relations*, contains the inner products of the normal vectors of the planar surfaces. A zero inner product means two perpendicular planes, while 1 or -1 means two parallel planes.

This group may also contain relations between a given vector $(\alpha_0, \beta_0, \gamma_0)$ and a set of planar normal vectors.

The fourth group, called the *line-line relations*, contains the inner products of the direction cosines of the lines.

The last group contains the inner products of the normal vectors of the planes and the direction cosines of the lines.

Partial Model Database

Using the described relation set, we can create the partial model for a given polyhedron building.

The simplest model is a cubic block. It contains 6 planes, 12 lines and 8 points. Figure 4(a) and (b) show the point-plane relations and the point-line relations.

The plane-plane, line-line and plane-line relations in the cubic model are illustrated in figure 5(a), (b), (c).

The second example is a hip roof model which has a roof with sloping ends and sides. It contains 9 planes, 17 lines and 10 points. Figure 6 shows the position relations in the model and figure 7 shows the angle relations.

In the real site model some buildings have very complicated structures. Dozens, or even hundred of linear objects may be involved in a complex building. We have manually labeled two site models (containing 57 buildings and 79 buildings respectively) and created a partial model database. In each of the site models, there are thousands of linear objects and more than ten thousand geometric relations.

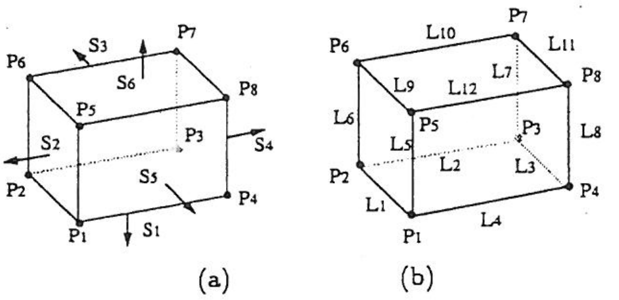

Figure 4: angle relations in flat roof model.
(a) point-plane relations, (b) point-line relations

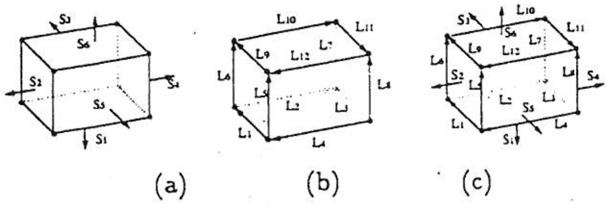

Figure 5: angle relations in flat roof model.
(a) plane-angle-plane, (b) line-angle-line, (c) plane-angle-line.

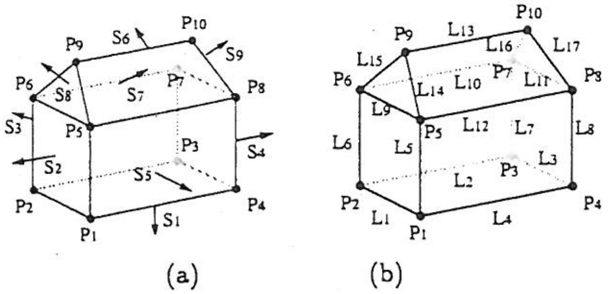

Figure 6: position relations in hip roof model.
(a) point-plane relations, (b) point-line relations

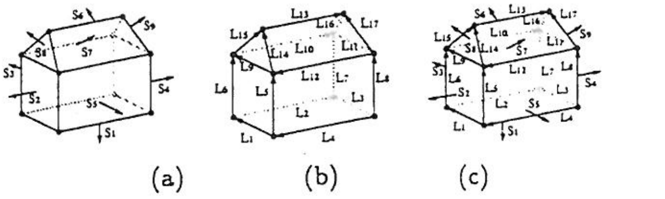

Figure 7: angle relations in hip roof model.
(a) plane-angle-plane, (b) line-angle-line, (c) plane-angle-line.

3.4.3 Constrained Optimization Framework

Having the partial object model and the perturbation model, we can modeling the estimation problem. Let $\Theta \in I\!R^m$ denote the parameters, $X' \in I\!R^m$ denote the observations, and $p(X' \mid \Theta)$ denote the likelihood function. In the building estimation problem, the parameters are the coordinates of the points, the normal vectors and distance constants of the planes, and the direction cosines and reference points of the lines.

Assume that the given optimality criterion is the maximum posterior probability, a Bayesian approach can be used to transform the problem into a maximum likelihood problem with constraints. Let the constraints be denoted by $\Theta \in C \subset I\!R^m$. The problem can be expressed as a constrained optimization problem.

$$\min\{-p(X' \mid \Theta) \mid \Theta \in C\}$$

The problem can be reformulated by taking log on the probability function. Under the assumption of Gaussian noise, we obtain a least squares model. The objective function is the sum of squared errors between the estimated point positions and the observed points.

$$\min_{\Theta} \quad \{f(\Theta) := (X' - X)^T \Sigma^{-1} (X' - X)\} \ .$$
$$\text{subject to} \quad \Theta \in C_\theta$$

where X denotes the unknown 3D points, and the feasible set C_θ is determined by the partial model and the unit length of the directional vectors.

If the noise effecting different 3D points is independent, the objective function can be rewritten as

$$f(\Theta) = \sum_{i=1}^{K}(\mathrm{x}_i' - \mathrm{x}_i)^T\Sigma_i^{-1}(\mathrm{x}_i' - \mathrm{x}_i)$$

where Σ_i is the covariance matrix of the ith point, and K is the number of observed points.

3.4.4 Building Parameter Data Files on CD-ROM

The CDROM contains the software for 3D optimization using the partial model, for partitioning the site model into building oriented models, and for preprocessing the input format and postprocessing the output data.

3.5 Map Labeling

The CDROM dataset contains map files for the visualization of the 3-D point placements. There are roof and ground maps, for model boards 1 and 2, prepared from points estimated with and without the partial building model constraints. The maps are given in both Postscript and Xfig formats. These maps each print to an 8x11 page with the labels having a font size of five. Figure 8 shows a detail of an output for MB1 and Figure 9 shows the same detail with constraints added. As shown by Figures 8 and 9, this provides a quick way visualize the effects of vision algorithms on 3-D data.

3.5.1 Problem Statement

Software for producing the labeled maps was nontrivial. To determine the placement of the labels, the software solves the following problem. Find the maximum size of a square such that each labeled point is the corner of exactly one square, and all squares are pairwise disjoint. These squares serve as a labeling space for each points.

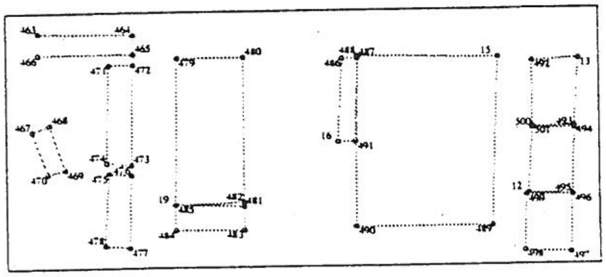

Figure 8: MB1 Roof Labeling Detail

A final processing step is performed that approximately maximizes the distance between labels without conflicts. The distance measures $D_1[n]$ and $D_2[n]$ are the distances from the center of point n's label to the center of the label's of the first and second nearest points. If at least one of $D_1[n]$ or $D_2[n]$ is $> 2\sigma$, then the label for that point n is examined. For all of these points n the best of the four label placements is chosen to minimize the function $D[n] = \frac{1}{1+D_1[n]} + \frac{1}{1+D_2[n]}$.

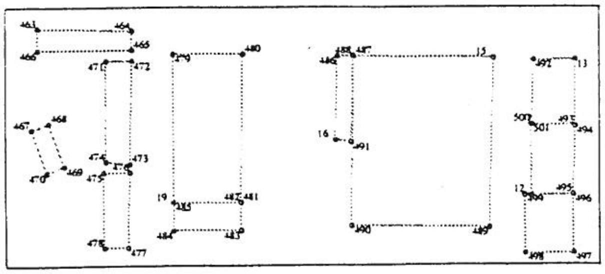

Figure 9: MB1 Roof Labeling Detail with Constraints

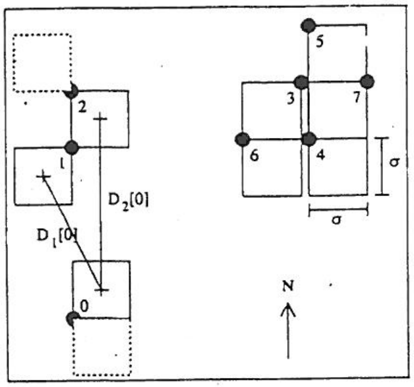

Figure 10: Example Map Labeling Solution

This final sweep changes the labels in Figure 10 on points 0 and 2. Point 5's label may better be placed in the NW corner, but more than one point within 2σ of point 5, computing the conflicts of changing the placement of point 5 will become computationally expensive.

Wolff's algorithm [5][1] finds an approximate solution to this NP-hard problem, as shown by the unshaded boxes in Figure 10. For for point 5 the original placement of Wolff's algorithm is not modified.

This code makes use of the Library of Efficient Data types and Algorithms (LEDA)[2]. LEDA is not in the public domain, but can be used freely for academic research and teaching (this does not include research within a company or state/government department).

4 Conclusion

This CDROM aerial model board imagery and its associated groundtruth can be used to characterize the performance of aerial image understanding algorithms. The performance of an algorithm based on this aerial imagry and groundtruth dataset can serve as a benchmark to the computer vision community. This common benchmark allows straightforward performance comparison between different groups' algorithms.

[1] http://www.inf.fu-berlin.de/labeling.html/

[2] http://www.mpi-sb.mpg.de/LEDA/leda.html

But more important than any benchmarking performance is the detailed testing and probing the data set will permit researchers to do with their algorithms. For when there is reasonable good ground truth, a deep exploring of why an algorithm sometimes gives answers far from the ground truth becomes possible.

References

[1] R.M. Haralick and L.G. Shapiro (1992), Computer and Robot Vision, Reading, MA:Addison-Wesley, 1992.

[2] R.M. Haralick, C.N. Lee, K. Ottenburg and M. Nolle", "Review and Analysis of Solutions of the Three-Point Perspective Pose Estimation Problem", International Journal of Computer Vision, Volume 13, Number 3, pp. 331-356, 1994.

[3] D. C. Nadadur, X. Zhang and R. M. Haralick. "Groundtruth Outline Drawing in Model-Board Images," Intelligent Systems Laboratory Technical Report No. ISL-TR-94-01, Updated: Feb 7, 1995.

[4] K. Thronton, D. C. Nadadur, V. Ramesh, X. Lui, X. Zhang, A. Bedekar, and R. M. Haralick. (1994), "Groundtruthing the RADIUS Model Board Imagery," Proceedings of the DARPA Image Understanding Workshop, 1994.

[5] F. Wagner and A. Wolff. (1995) "An Efficient and Effective Approximation Algorithm for the Map Labeling Problem," Proceedings of the 3rd Annual European Symposium on Algorithms, 1995.

[6] K. Mehlhorn and S. Näher. (1995) "LEDA: a platform for combinatorial and geometric computing," Communications of the ACM Vol.38 pp.96-102, 1995.

[7] A. Bedekar, and R. M. Haralick. "A Bayesian Method for Triangulation and its Application to Finding Corresponding Points," ISIP94.

[8] A. Bedekar. "Finding corresponding points based on Bayesian triangulation," Master's thesis, Dept. of Electrical Engg., Univeristy of Washington, 1995.

[9] X. Zhang, R.M. Haralick, V. Ramesh, and A. Bedekar. "A Bayesian Corner Detector: Theory and Performance Evaluation," Image Understanding Workshop, 1994.

[10] O.D. Fageras. "Three-Dimensional Computer Vision: A Geometric Viewpoint", MIT Press, 1993.