

Automatic Description of Gable Buildings of Polygonal Boundaries with Multiple Images¹

ZuWhan Kim, Andres Huertas, and Ramakant Nevatia
Institute of Robotics and Intelligent Systems
University of Southern California
PHE 204, MC-0273 Los Angeles, CA 90089-0273
{zuwhan, huertas, nevatia}@iris.usc.edu

Abstract

We present a model-based approach to detecting and describing composition of buildings with flat or gabled polygonal rooftops. Previous approaches have dealt with either simpler models or models which lack geometric information. In spite of increasing model complexity, we maintain the computation affordable by effectively using multiple overlapping images. We obtain rooftop hypotheses in 3-D by using 3-D lines and junctions generated from multiple images. Image-derived unedited elevation data is used to assist feature matching, and to generate rough cues of the presence of 3-D structures. Experimental results are shown on complex buildings.

1. Introduction

Three-D object description is a key task of computer vision. One practical application for the 3-D object description problem is that of building detection and description from aerial images, which has been an active research area [1]. It can greatly improve the automation of 2-D or 3-D map generation which can be used in various applications including radiowave reachability tests for wireless communications, computer graphics, virtual reality, and mission planning.

A single intensity image has been used for early building description systems [2], [3], [4]. However, in general, multiple aerial images can be obtained with small extra cost. Most of the recent work in building detection has focused on the stereo or multi-view analysis [5], [6], [7], [8], [9], where 3-D information is obtained simply by matching features such as pixels, lines and junctions. Range data is also used as an important cue [10], [11], [9].

As in other 3-D object description problems, model representation of buildings takes an important role. The

complexities of the buildings that can be described vary from that of hexahedral buildings or their compositions to any polyhedral buildings. Extrusions of rectangular rooftops are used to represent hexahedral buildings [4], [6], and refined polygonal meshes are used for polyhedral ones [8], [9]. Another axis of the representation is the level of geometric information. For example, when a hexahedral building is represented as an extrusion of rectangular rooftops, we can infer geometric information, such as whether a certain 3-D point belongs to a rooftop boundary or wall vertical, directly from the representation. On the other hand, unrestricted polygonal meshes [8] lack such geometric information; therefore it is more difficult to use the result in this form for higher-level processing, such as building identification.

In this paper, we use a representation which contains high-level geometric information as well as being able to represent complex buildings - compositions of flat or gable buildings of polygonal boundaries. A building component with a flat polygonal rooftop can simply be represented as an extrusion of the rooftop points. The representation of a gable rooftop is illustrated in Figure 1. It consists of a list of rooftop boundary points and a list of spine points as well as the relationships between rooftop corner points and gable spines (*spinal relationships*).

We have introduced a basic framework and preliminary results in [12]. We maintain computation affordable by effectively using multiple (more than 3) overlapping images and image-derived unedited digital elevation models (DEM's). Example input images of a building from three different views are shown in Figure 2a,b,c. Figure 2d is an image-derived unedited DEM which will be used as additional input. Note that, in contrast with [10] and [9], these DEM's are not accurate enough to retrieve building model directly from it. Although DEM's can be computed from high resolution (sub meter) images, the underlying correlation methods used have inherent limitations and

1. This research was supported by a MURI subgrant from Purdue University under Army Research Office grant No. DAAH04-96-1-0444.

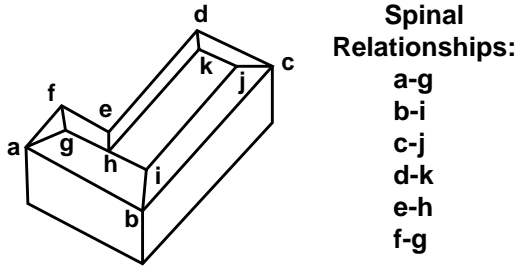


Figure 1. Representation of a gable building. A list of rooftop points, (a b c d e f), and a list of spine points, (g i j k h), as well as their *spinal relationships* are stored.

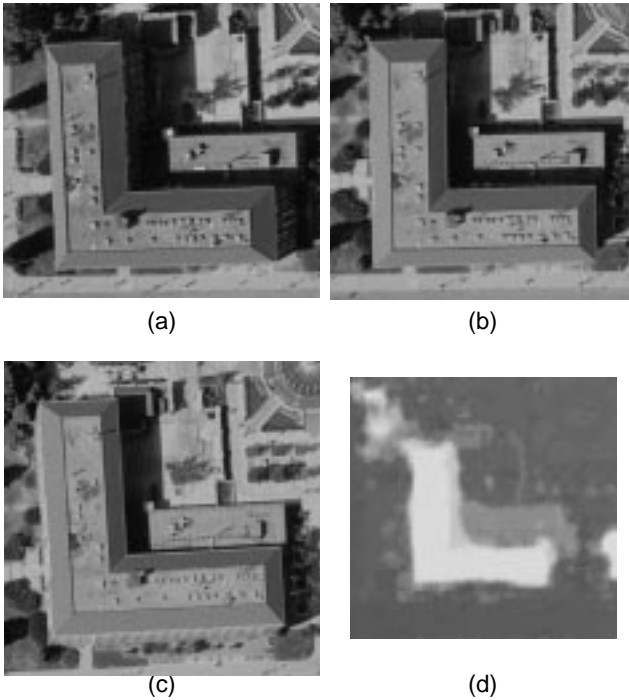


Figure 2. (a, b, c) Example aerial images of a gable building (0.25m/pixel resolution); and (d) an image-derived unedited DEM (0.5m/pixel resolution)

produce errors at and near building (and other) depth discontinuities. Research to increase the quality of DEM using more than 10 images is found in [13].

Figure 3 shows a flow diagram of our method. The system uses multiple images at about 1 meter resolution and an unedited DEM at a similar resolution. Note that the rough DEM is used primarily to provide cues that help reduce the search spaces and validate feature matches. First, 2-D features, lines, junctions and parallel relationships, are extracted from the images. Next, we derive 3-D features from groups of matched 2-D features over multiple views to generate rooftop hypotheses. All 2-D features may not be present in all views, therefore, to generate 3-D

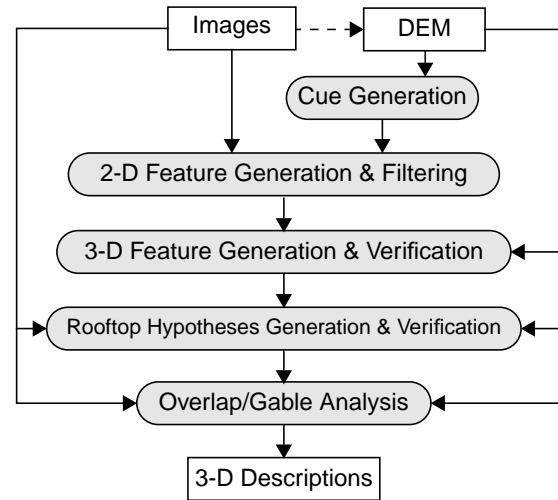


Figure 3. A flow diagram of the suggested approach.

features, pairwise feature matching across all views is performed first, followed by grouping of matched pairs.

The next step is rooftop hypotheses generation. Flat polygonal rooftop boundary hypotheses are generated by neighborhood searches on 3-D features. We use level-of-detail technique to reduce time. Then, generated hypotheses are verified with *expandable Bayesian networks* [14]. Finally, overlap analysis is performed on the generated hypotheses and a gable analysis is applied to give final building descriptions.

Section 2 describes 2-D feature extraction and filtering focusing on the use of DEM information. Section 3 deals with the use of multiple images to generate and verify 3-D features. In Section 4, we describe the rooftop hypothesis generation, and hypothesis verification and overlap analysis are described in Section 5. The gable analysis is shown in Section 6 and, in Section 7, time complexity is briefly analysed. Experimental results are shown in Section 8, and the conclusion is given in Section 9.

2. 2-D Feature Extraction and Filtering

2-D Feature Extraction. First step is to extract line features from images. Extracted line segments are grouped into “linear” features by collapsing collinear lines of small gaps. The junctions among these linears represent strong point features that are well localized and have reduced ambiguity. Thus, we use junctions as an important feature. Junctions are extracted by grouping nearby linears which make angles bigger than 60° . T-junctions are also extracted. A T-junction is considered to be two L-junctions for further processing. Building structures exhibit a great

deal of parallelism in their design and construction. We extract parallel relationships from each view.

Line Filtering with DEM Cues. The 1316 linears features extracted from Figure 2a are shown in Figure 4a. We see large numbers of distracting linears from trees, roads and other structures nearby, which make the computation significantly large. We use DEM cues to eliminate these distracting linears. Although DEM data does not give explicit model, it still gives a rough idea of where the buildings are located. We follow an approach of [11] to generate rough cues from a DEM image. The DEM image (Figure 2d) is first convolved with a Laplacian-of-Gaussian filter to smooth the image and locates the object boundaries. Then the positive-valued regions bounded by the zero-crossings in the convolution output is extracted. Figure 4b shows cues generated from the DEM image of Figure 2d.

Once rough building cues are generated, the linears far from the cues are eliminated first. Then, the linears of “inappropriate angles” are eliminated. Note that the boundary lines of (polygonal) building rooftops are mostly of two or three dominant angles. We generate a length-weighted histogram of line angles from the linears of each cue and use only linears near the *dominant angles* for the rooftop boundary hypotheses generation. The *dominant angles* are determined by choosing the two highest peaks in the histogram. The resulting filtered linears are shown in Figure 5a. Only 320 of 1726 linears remained. Figure 5b shows 128 junctions grouped from the filtered linears.

3. 3-D Feature Generation and Verification

We use two types of 3-D features; 3-D linears and 3-D junctions. A 3-D linear is a group of matched linears (“*member linears*”) and a 3-D junction is a group of matched junctions (“*member junctions*”). We only consider flat (parallel to the ground) rooftop boundary hypotheses. Therefore, an assumption is applied that these 3-D features are parallel to the ground. With this assumption, we can easily reconstruct the 3-D geometry of a 3-D feature once its height is given.

3-D Feature Generation. We generate 3-D features by merging pair-wise matches of 2-D features. The pair-wise matches are generated by using epipolar geometry (camera parameters are known). Given a matched pair, 2-D features are collected from other views which have compatible orientations and heights. The height of a 3-D feature is estimated by combining pair-wise height estimates.

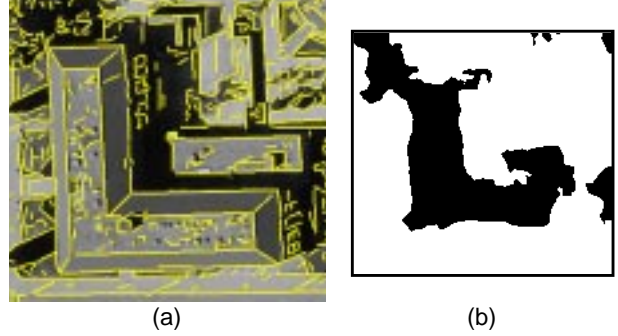


Figure 4. “Linear” features extracted from Figure 2a. 1726 linears were extracted

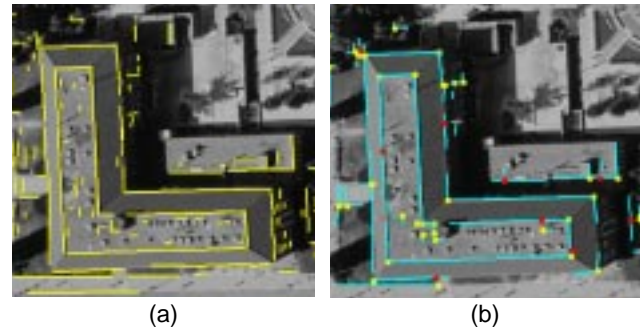


Figure 5. (a) 320 linears filtered by location and angles; and (b) 128 junctions grouped from the filtered linears. Red ones are T-junctions.

In [12], we have introduced the *epipolar alignment* problem. When linears (from different views) of a match are aligned to the epipolar lines, the estimated height may contain a large error while the error will be smaller when they are perpendicular to the epipolar line. Due to the epipolar alignment (or near-alignment), the errors in height estimations will be of various degrees according to the angles of linears and camera parameters. Therefore, how to combine these pair-wise estimates of various degrees of errors is not clear. We also face similar problems with junction matches. Although matching junctions does not suffer from the epipolar alignment problem, the accuracy of height estimation depends on the physical distance of the cameras where the images were taken from (baseline length).

We use a probabilistic approach [12], where, the height estimates are represented as Gaussian random variables. Given a line pair, the height (mean of the Gaussian random variable) is estimated from stereo analysis, and a confidence interval (standard deviation) is obtained assuming possible displacement errors of lines in the image space (by assuming 2.5 pixel location error). Based on this formalism, we estimate the height \hat{h} and confidence interval σ of a 3-D linear from n pair-wise height estimates, $(h_1, \sigma_1), \dots, (h_n, \sigma_n)$;

$$\hat{h} = \left(\sum_i \frac{h_i}{\sigma_i^2} \right) / \left(\sum_i \frac{1}{\sigma_i^2} \right) \quad \sigma^2 = 1 / \left(\sum_i \frac{1}{\sigma_i^2} \right). \quad (1)$$

We also use probabilistic reasoning for height compatibility tests in hypothesis generation. Two height estimates, (h_1, σ_1) and (h_2, σ_2) , are considered to come from the same height distribution when $\mu - \sigma \leq 0 \leq \mu + \sigma$, where $\mu = h_1 - h_2$ and $\sigma^2 = \sigma_1^2 + \sigma_2^2$. Derivations of the above equations can be found in [12].

3-D Feature Verification with DEM. The 163 3-D linears generated from five images of a building in Figure 2 is shown in Figure 6a. Despite that these 3-D linears have *members* in at least 3 different images, still a large number of ambiguities exists due to distracting parallel lines and the epipolar alignment problem. Thus, we use DEM data (Figure 2d) to eliminate some false matches. To verify a 3-D linear, it is projected onto the DEM image. The projected line is used to compute statistics of the DEM values in regions (sampling windows) adjacent and on both sides of the projected line. A 3-D linear is verified when its height is compatible to the DEM statistics. A 3-D junction is verified when both of its branches are verified. Figure 6b shows verified 3-D linears. Note that wall base lines also remain as well as rooftop boundary lines since they are from the real 3-D lines. Among the 412 3-D linears (Figure 6a) only 94 of them were verified.

Finally, *polarities* are assigned to 3-D linears. The polarity of a 3-D linear is defined as positive when its height is the same as that of its left-hand side. Three-D junctions are verified when both of their branches are verified. The polarity of a 3-D junction is positive when its height is the same as that of its inner-side. By knowing that which side of a 3-D feature a building lies, we can reduce the search time of the hypotheses generation significantly. Figure 7 shows the 2-D features of an image which are the members of the 3-D verified features. Polarities are shown with arrows. Twenty eight from 100 3-D junctions were verified.

4. Hypotheses Generation

Rooftop boundary hypotheses are generated by a neighborhood search on 3-D features. However, we still see clouds of 3-D linears in Figure 6b which may increase the search complexity exponentially. To reduce the search complexity, we apply a level-of-detail technique. First, coarse-level 3-D features are generated by merging nearby 3-D features. The neighborhood search is performed on the coarse-level 3-D features and a relaxation procedure is applied to refine the generated “coarse hypotheses.”

Three-D coarse linears are generated by grouping 3-D linears of similar heights and locations. Three-D coarse

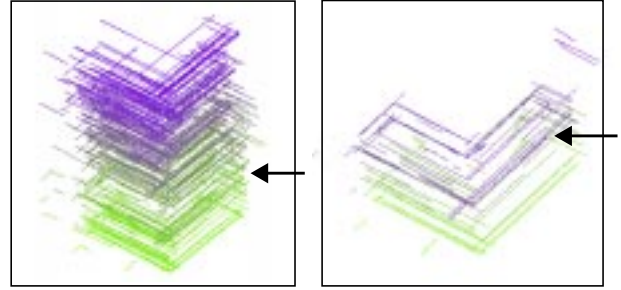


Figure 6. (a) The 163 3-D linears generated from five images of a building in Figure 2. Purple colored ones are higher than greens. The arrow shows the actual height of a building. (b) 3-D linears verified with DEM. The arrow shows the actual height of a building. Many false matches were eliminated.

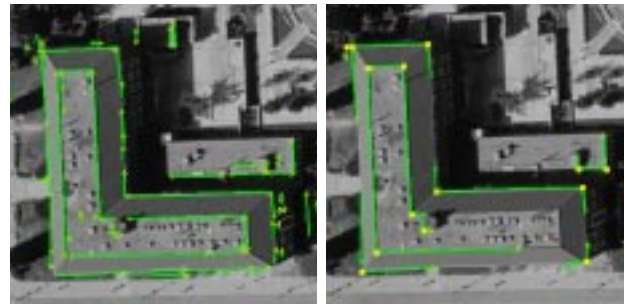


Figure 7. 2-D features which belong to the verified 3-D features. Polarities are shown with arrows.

junctions are generated by grouping 3-D junctions where their branches (3-D linears) were grouped into common 3-D coarse linears. Figure 8 shows the 52 3-D coarse linears generated from the 94 3-D linears of Figure 6b. We see that most of the ambiguities are removed. From the same example, 12 3-D coarse junctions were generated from 28 3-D junctions.

Once 3-D coarse features are generated, a search is performed on these features to group them into *coarse* hypotheses. A depth-first search starting from a 3-D coarse junction is performed on *neighborhood features*. Neighborhood relationships are defined by the branch relationship of 3-D coarse junctions and linears. Since polarities are defined for all the 3-D features we only generate counter-clockwise rooftop hypotheses.

However, grouping only neighbor features does not generate many of the desired hypotheses since some features may be missing. Therefore, we also use parallel relationships; two hypotheses of a compatible height, which have linears that are parallel to each other, are combined once again to make another hypothesis. Note that the par-

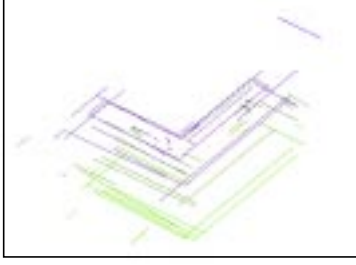


Figure 8. The 44 3-D coarse linear generated from the 94 3-D linear of Figure 6b.

allel relationships are not used in the neighborhood search because they increase the branching factor enormously.

The next step is to refine hypotheses generated from 3-D coarse features. Given a hypothesis, among the “member” 3-D features, x_1, \dots, x_n , of a 3-D coarse feature, the one with the best supporting score is chosen, where the supporting score, $s(x_i)$, of the i -th 3-D feature, x_i , is defined by the following equation:

$$s(x_i) = \alpha s_L(x_i) + \beta s_N(x_i), \quad (2)$$

where $s_L(x)$ is a local score, $s_N(x)$ is a neighborhood score, and α and β are the weights for these scores (constants).

The local score, $s_L(x)$, of a 3-D feature x is a combination of the *strength* (intensity contrast of sides) of the “member” 2-D linear and the coverage of actual 2-D linear of the corresponding building side. The neighborhood score, $s_N(x)$, is given by the following equation:

$$s_N(x) = \sum_j s(b_j)c(x, b_j) + \sum_k s(f_k)c(x, f_k), \quad (3)$$

where b_j is the j -th 3-D feature of the backward 3-D coarse feature, f_k is the k -th 3-D feature of the forward 3-D coarse feature, and $c(x, y)$ is the compatibility function of 3-D features x and y . The compatibility function, $c(x, y)$, of 3-D features x and y is based on the distance between their 2-D linear (or branches of 2-D junctions).

Note that we need to obtain $s(b_j)$ and $s(f_k)$ for all j 's and k 's to get $s_N(x_i)$, which requires the values of $s_N(b_j)$ and $s_N(f_k)$ Eq (2). Therefore, an iterative relaxation technique is applied. Initially, $s(x)$ is set to $s_L(x)$ for all the 3-D features belong to the coarse hypothesis. Then, for each iteration, Eq (3) and Eq (2) are repeatedly applied to get $s(x)$. After a certain number of iterations (3 in ABERS), a 3-D feature, x_i , of the highest score, $s(x_i)$, is chosen among the member 3-D features of a 3-D coarse feature.

The generated and refined hypotheses are collections of 3-D features. We need to obtain 3-D rooftop boundaries from them. The 3-D positions of 3-D junctions (Section 3) are used to determine the corner points of the rooftop

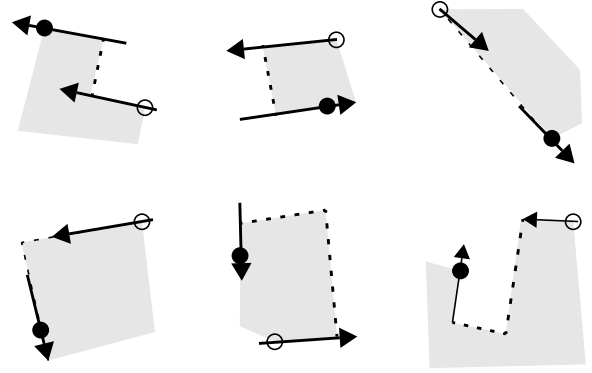


Figure 9. Suggested closures (dashed lines) for various alignments of 3-D linear. Arrows represent 3-D linear with polarities. Blank circles represent the previous corners and filled circles represent the next corners. Buildings are regarded to lie on the shaded sides.

boundaries. However, most of the generated hypotheses are not *closed*; they are mostly chains of 3-D features or pairs of the chains (when parallel relationships are applied). Hence, we need to determine proper closures of them. Note that junctions always have branches (3-D linear) as neighbors. Thus, the chains always end with 3-D linear, and closures are made from them. Figure 9 shows possible alignments of 3-D linear (arrows) and suggested closures [12]. We only consider closures parallel or perpendicular to those 3-D linear. Note that there are more than one pair of end-points for a 3-D linear. Closures are generated from all the possible end-points and the one with the best *line support* (coverage of the supporting line segments and the strength of the distracting line segments) is chosen.

5. Hypotheses Verification and Overlap Analysis

Once rooftop hypotheses are obtained, supporting evidence is collected for the hypotheses. Evidence support consists of line support, wall vertical line support, and darkness of the shadow region. Line support consists of the coverage of the supporting line segments (**RP**) and the strength of the distracting line segments (**RN**). Wall vertical line support (**WV**) is the coverage of the supporting line segments for the possible wall verticals (building corners). We only consider geometrically and photometrically visible lines, which means that self-occluded lines and lines in the shadow side of a building are not considered. The darkness of the possible shadow region (**SD**) is the percentage of dark pixels in the possible shadow region.

Note that the computation for some of the evidence, such as the darkness of shadow region (**SD**), is relatively

expensive than others. To reduce the computation, we apply a filtering procedure; in the first stage (*hypotheses selection*) hypotheses with bad roof line support (calculated from **RP** and **RN**) are filtered out, and the rest of the evidence is collected and applied for the remaining hypotheses. Note that the number of images may vary at runtime. To compute $P(\mathbf{Building}|\mathbf{Evidence})$ with varying number of images, *expandable Bayesian networks* (EBN) [14] are applied. For image-derived evidence variables, EBN provides *repeatable* nodes, which instantiate (make copies) at runtime. A simple EBN for the hypotheses selection is shown in Figure 10a and the one using full evidence is shown in Figure 10b. Additional information was used, such as the size (area) of the projected rooftop hypotheses on an image (**Size**) and projected length of wall verticals (**WL**). All the continuous evidence variables were discretized into 5 levels, a binary node was used for **Size**, and a ternary node was used for **WL**.

It is common that more than one hypotheses are verified for a single building component, where these hypotheses represent parts of an actual building. Therefore, we need to choose the best possible building component. However, comparing two verified hypotheses according to their verification score, $P(\mathbf{Building}|\mathbf{Evidence})$ of the EBN in Figure 10b, is not an accurate way because that binary classifier is not designed and learned to compare two good building hypotheses but to determine whether a certain hypothesis is building or not. Therefore, we need a *comparative classifier*, which takes two sets of evidence variables as input and determines the probability that one hypothesis is better than another.

For the *comparative classification*, we use the vector difference between two sets of evidence variables since all the evidence variables are continuous. Another EBN, which takes the vector differences as input, was designed and learned for the overlap analysis. The structure of the *comparative* EBN is the same as the one for the hypothesis verification (Figure 10b) while the parameters are totally different.

The learning dataset for the comparative classifier is obtained by specifying *accurate* and *inaccurate* hypotheses by hand. Then, for all the overlapping pairs of accurate and inaccurate hypotheses, the learning data is obtained by taking the vector difference of their evidence variables. Note that the classification result should be *commutative*; $C(v) = C(-v)$, where v is the vector difference between two sets of evidence comparing, and $C(v)$ is the classification result for v . Therefore, for each evidence pairs, two learning data, v and $-v$, are generated.

The overlap analysis procedure follows the following iterative steps. First, one of the rooftop hypotheses which is superior to all of its overlapping hypotheses is selected, and all of its overlapping hypotheses are eliminated. Then,

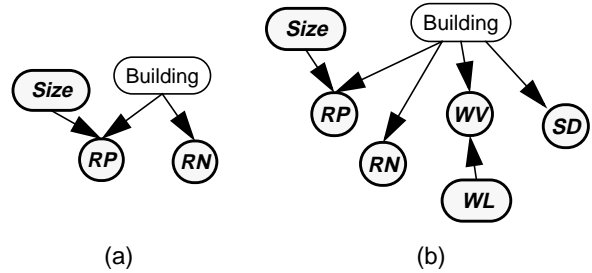


Figure 10. EBN's used for (a) hypotheses selection; and (b) hypotheses verification with full evidence.

the same procedure is repeated for the remaining hypotheses until no more hypothesis remains.

6. Gable Analysis

The next step is that of gable analysis. For each corner of a rooftop boundary hypothesis possible *gable support lines* (diagonal lines from the corners) are gathered. Figure 11 shows the search windows (red) and the gable support lines found for an example hypothesis. Once gable support lines are collected from all the images, 3-D linears are generated by matching them. Note that these 3-D linears are *not* parallel to the ground. Line matching without any restriction is very susceptible to noise. However, fortunately, we know that one of the 3-D end-points of the 3-D linear corresponds to a corner point of the rooftop hypothesis. Figure 12 illustrates the situation. Let lines l_1 and l_2 are the projections of a gable support line l (for a corner point p) to view₁ and view₂. The problem is to reconstruct the 3-D line l from the line segments (q, q') and (r, r') on l_1 and l_2 . Since q is the projection of a point on l , the 3-D line l intersects with the line of sight l_q for q . Knowing that l passes through p , we find one constraint that l lies on the plane Π_q which contains both l_q and p . In the same manner, l lies on the plane Π_r determined by l_r , the line of sight for r , and p . Thus, l is the intersection of Π_q and Π_r (unless Π_q and Π_r are identical which is rare for aerial images). Since we know that l passes through p , we only need to know the 3-D orientation of it, which is obtained by cross-producting the normals of Π_q and Π_r . The normal of Π_q (or Π_r) is obtained by cross-producting l_q (l_r) and (p, x) where x is any point on l_q (l_r).

When 3-D gable support linears are found for most of the corner points, a search is performed to find spines. Example search results for a building corner p is illustrated in Figure 12. First, 2-D linears parallel to the neighboring building sides are gathered. When gable support linears are available, we can make relatively smaller search windows from the 3-D end-points of the linears as shown in Figure 12 (yellow box). The search results are shown in



Figure 11. Gable support lines (shown in green). Red boxes are search windows.

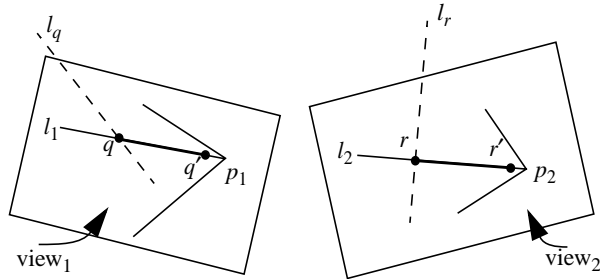


Figure 12. Determining the 3-D orientation of a gable support linear.

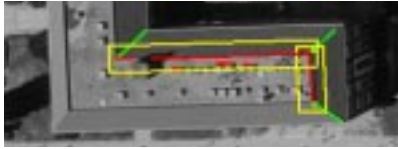


Figure 13. Searches for spines. Gable support linears are shown in green, search windows are in yellow, and the red line segments are the resulting 2-D linears.

red. The spine candidates are the 3-D linears of reasonable height which have those 2-D linears as their members. The spine corner point corresponding to the boundary corner point p will be the intersection of a pair of the candidate 3-D linears. For all the candidate pairs, local *line support* (Section 4) and compatibilities with candidate pairs of neighbor corners are calculated, and the best pair for each corner is determined. The spine corner point is set from these pairs.

A gable corner point can also be determined by a single spine, not a pair; for example, the corner point \mathbf{g} of Figure 1. In such a case, the corner point is determined by the intersection of the two gable support linears ($\overline{\mathbf{ag}}$ and $\overline{\mathbf{fg}}$ in Figure 1) or the intersection of a gable support linear and the spine linear. Not all the spine corner points can be found from the above method. For a missing spine corner, a corner hypothesis is generated from the neighbor corner points forcing the parallel relationship. Such a corner

hypothesis is verified with the *line support*. Finally, a gable is generated by collecting all the spine corners.

7. Time Complexity

The time complexity of 3-D feature generation is $O(ln^2)$, where l is the average number of 2-D linears per image, and n is the number of images. The number of 3-D coarse linears is bound by $O(l)$ and the number of coarse hypothesis, before parallel relationships are applied, is $O(lb^k)$, where k is the maximum number of rooftop corners, and b is the *branching factor* of the graphs. Although $b = O(l)$, in the worst case, the actual number of the coarse hypothesis is usually small (smaller than three times the number of coarse junctions, in most cases) because of broken linears and missing junctions. Also, when we limit the size of DEM cues, b^k is bound by $O(1)$. The total number of hypotheses generated are $O(l^2b^{2k})$ when parallel relationships are applied. However, when we limit the maximum width of a building, it can be reduced to $O(lb^k)$. Hence, the total time complexity of ABERS is $O(ln^2) + O(lb^k)$.

8. Experimental Results

Figure 14 is the result for the example building of Figure 2. Five images of the 406.8×442.8 average resolution were used. Total time spent was 4 minutes and 58 seconds. We see that a near-perfect building description is obtained in spite of broken roof boundary lines (due to accidental illumination) and distracting lines (Figure 4).

Another result is shown in Figure 15a. Nineteen minutes and 15 seconds were spent to process five images of 500.6×552.2 average resolution. Figure 15b shows the extracted line segments from an image, which shows the complexity of the problem. We see broken boundary lines and large numbers of distracting lines both on the rooftop and the ground. Figure 16 shows more results on gable buildings.

9. Conclusion

3-D object description is a difficult problem especially when the target objects have complex shapes. The computation to get reasonable detection may increase exponentially when the model complexity increases. This paper suggest an approach to combine diverse information effectively to control such increasing computation. Experimental results show that the suggested approach is promising.



Figure 14. A detection result for the example building of Figure 2.

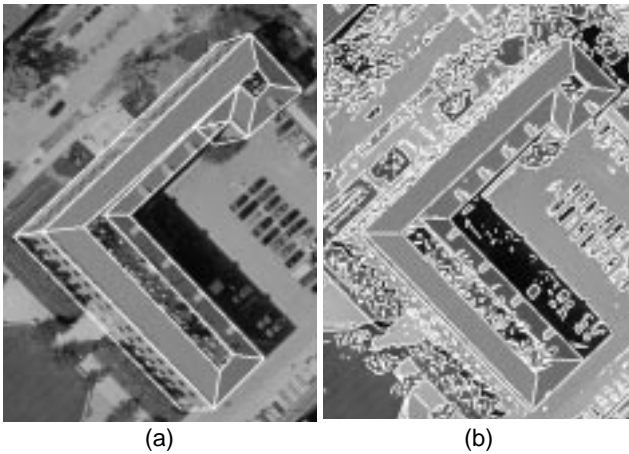


Figure 15. (a) A detection result for another gable building; and (b) the line segments show the difficulty of the problem.

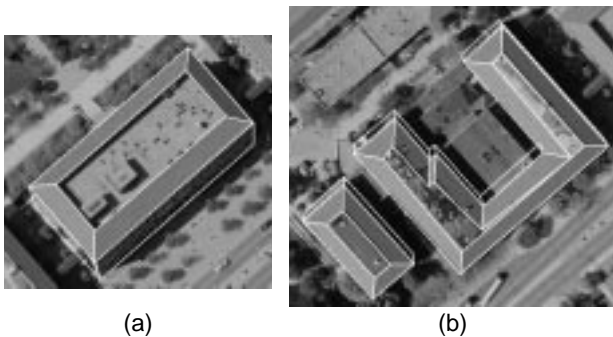


Figure 16. More results on gable buildings.

References

[1] A. Gruen and R. Nevatia (editors), *Computer Vision and Image Understanding: Special Issue on Automatic Building Extraction from Aerial Images*, Vol. 72, No. 2, 1998.

[2] A. Huertas and R. Nevatia, "Detecting Buildings in Aerial Images," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 131-152, 1988.

[3] R. Irving and D. McKeown, "Methods for Exploiting the Relationship Between Buildings and Their Shadows in Aerial Imagery," *IEEE Trans. Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1564-1575, November/December 1989.

[4] C. Lin and R. Nevatia, "Building Detection and Description from a Single Intensity Image," *Computer Vision and Image Understanding*, vol. 72, no. 2, 1998.

[5] M. Roux and D. McKeown, "Feature Matching for Building Extraction from Multiple Views," *Proc. IEEE Computer Vision and Pattern Recognition*, Seattle, WA, pp. 46-53, 1994.

[6] S. Noronha and R. Nevatia, "Detection and Description of Buildings from Multiple Aerial Images," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.588-594, 1997.

[7] R. Collins, C. Jaynes, Y.-Q. Cheng, X. Wang, F. Stolle, E. Riseman and A. Hanson, "The Ascender System: Automated Site Modeling from Multiple Aerial Images," *Computer Vision and Image Understanding*, vol. 72, no. 2, pp. 143-162, 1998.

[8] C. Baillard and A. Zisserman, "Automatic Reconstruction of Piecewise Planar Models from Multiple Views," *Proc. IEEE Computer vision and Pattern Recognition*, Fort Collins, Colorado, pp. 559-565, June, 1999.

[9] B. Ameri, "Feature Based Model Verification (FBMV): a New Concept for Validation in Building Reconstruction," *Proc. ISPRS Congress*, pp. 24-35, 2000.

[10] M. Cord, M. Jordan, J.-P. Cocquerez and N. Paparoditis, "Automatic Extraction and Modelling of Urban Buildings from High Resolution Aerial Images," *Proc. ISPRS Automatic Extraction of GIS Objects from Digital Imagery*, Munich, Germany, pp. 187-192, September, 1999.

[11] A. Huertas, Z. Kim, and R. Nevatia, "Use of Cues from Range Data for Building Modeling," *Proc. DARPA Image Understanding Workshop*, pp. 577-582, 1998.

[12] Z. Kim, A. Huertas, and R. Nevatia, "Automatic Description of Complex Buildings with Multiple Images," *Proc. 5th IEEE Workshop on Applications of Computer Vision*, pp. 155-162, 2000.

[13] C. Vestri and F. Devernay, "Improving Correlation-based DEMs by Image Warping and Facade Correlation," *Proc. IEEE Computer Vision and Pattern Recognition*, vol. 1, pp. 438-443, 2000.

[14] Z. Kim and R. Nevatia, "Learning Bayesian Networks for Diverse and Varying Numbers of Evidence Sets," *Proc. Int'l Conf. on Machine Learning*, pp. 479-486, 2000.