Finding Planar Regions in Noisy 3D Grid Point Data

Justus H. Piater and Edward M. Riseman

CMPSCI Technical Report 96-47

November 1996

Computer Science Department
Lederle Graduate Research Center
University of Massachusetts
Amherst, MA 01003-4601

{piater|riseman}@cs.umass.edu

Finding Planar Regions in Noisy 3D Grid Point Data

Justus H. Piater and Edward M. Riseman

Abstract— This paper describes an algorithm for finding regions of planar surfaces in data given as three-dimensional points located in a (possibly incomplete) grid. It was motivated by noisy range data obtained from IFSAR or stereo correlation. The procedure basically consists of the following steps: (a) fitting of robust planar patches, (b) region merging, (c) region refinement, (d) region splitting, and (e) boundary finding. The steps (b)–(d) are iterated until a convergence criterion is met.

A major design principle was the lack of assumptions about specific characteristics of the input data. Hence, heuristic parameters common in most of today's range image segmenters were avoided where possible. Rather, the same fit criteria are applied at all steps of the procedure. The algorithm can incorporate information from optical imagery where available, and cooperate with higher-level site reconstruction tasks. However, a major drawback is the resultant computational complexity of the procedure, which becomes significant especially where planar regions grow very large. Therefore, some optional heuristic parameters are provided for speedup, without degrading the results. This paper describes the algorithm, illustrates some results on IFSAR, digital elevation maps (DEMs) from aerial stereo images, and range data, and discusses advantages as well as limitations.

1 Introduction

Localization of planar regions in 3D data is a fundamental, yet difficult problem with many applications in object recognition, target detection, and terrain reconstruction. Traditionally, important cues are taken from optical imagery registered to the 3D data. In site reconstruction from aerial images, for example, potentially planar regions are segmented in optical images, and are tested for planarity in noisy 3D interferometric synthetic aperture radar (IFSAR) data by fitting planes to the corresponding regions in 3D (Jaynes et al. 1994). However, this (top-down) approach fails where no data sources other than the 3D data exist, or where the information they provide is misleading or incomplete. The motivation for this research was the need (a) for data-driven support where optical imagery is available but does not provide sufficient information for 3D reconstruction, and (b) for an entirely data-driven (bottom-up) approach for finding planar regions, without such additional information. Although many applications are possible, this paper mainly addresses the issue of site reconstruction from 3D elevation maps obtained by IFSAR or by stereo matching of optical images. We note that IFSAR data and processing is predicted to become a very heavily used source of information in the near future.

In IFSAR, the complex phase difference of two SAR images is used to derive a height measurement at each image location (see e.g. Giglio and Carlisle (1995) for a concise introduction). Typically, the SAR images are acquired using two antennas mounted on a single aircraft. Alternatively, SAR data from two passes of aircraft equipped with a single SAR system can be registered using highly accurate differential GPS information. The resultant digital elevation maps (DEM) tend to be speckled with non-Gaussian noise and contain many dropouts caused by radar shadows (Fig. 1).

The quality of depth reconstruction from stereo largely depends on the reconstructed objects and the matching algorithm employed. Here we are concerned with DEM's generated by the UMass terrain reconstruction system (Schultz 1995). It performs multi-resolution stereo correlation matching in aerial images without prior feature extraction. The images may be nadir views, but the system works effectively even on highly oblique views from very different viewing angles. Accurate elevation maps can be expected where detailed and structured texture is available, whereas lack of texture may cause stereo mismatches, which in turn produce irregularities in the output DEM. On the other hand, the DEM's are relatively free from random uncorrelated noise. A

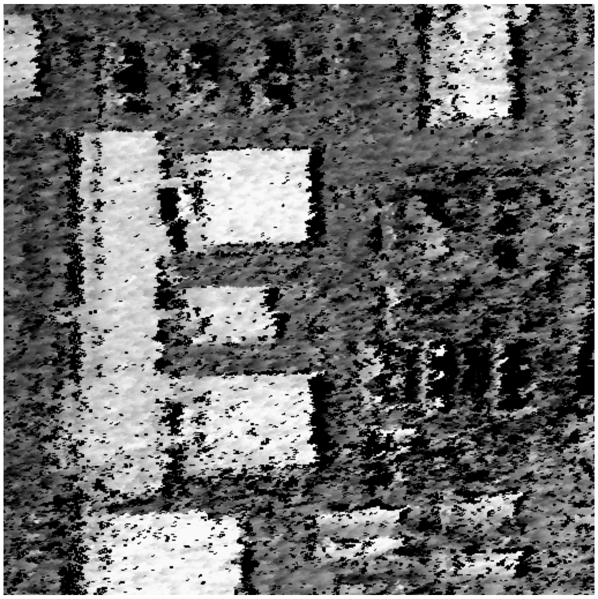

Figure 1: IFSAR data from Kirtland AFB. Gray values represent heights, and black are dropouts (i.e. points without height information). Note the speckled noise and the clustered structure of the dropouts.

striking characteristic of errors is that sharp changes in elevation often appear rounded by smoothing effects induced by the hierarchical matching technique, where large correlation masks had to be employed (Fig. 2). Dropouts do not generally occur, but low confidence on mismatched points that are occluded in one of the images serve as the equivalent.

This paper describes an algorithm that was designed to perform planar region extraction on difficult types of input data, where the difficulty is given by extreme noisiness and/or lack of 3D edges. It can operate as an independent, entirely data-driven tool, or it can be used to aid a higher-level reconstruction procedure. The overall problem addressed in this paper can be characterized as follows: A set of 3D points is given as input, arranged in a rectangular (i, j) grid topology (such that each point can be indexed by its (i, j) coordinates) which is assumed to form a surface. Note that the grid need not be regularly spaced. It is only needed to provide neighborhood relationships. For example, it can be given by the image coordinates of an oblique view, or by the raster of a range image. Furthermore, the grid may be incomplete, i.e. points may be missing. The goal is to determine connected sets of points forming planar regions in this surface. For use by a higher-level procedure and for visualization purposes, the boundaries of each region are extracted as lists of 3D points. This includes both exterior (hull) and interior (hole) boundaries.

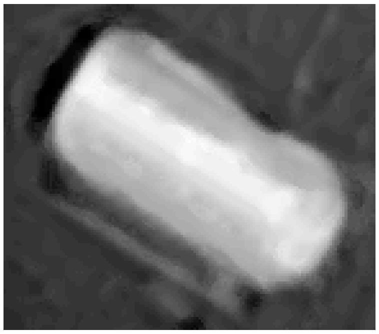

Figure 2: DEM generated via stereo correlation from two optical images. The scene contains a peaked-roof building. Note that most edges appear smoothed.

This problem is somewhat related to range image segmentation (RIS). Two dominating approaches in the literature are (1) parameter clustering or region growing (e.g. Kaveti et al. 1994, Krishnapuram and Freg 1992) and (2) patch fit/merge algorithms (e.g. Jones and Illingworth 1994, Hu and Mehrotra 1992, Boulanger and Godin 1992). Note, however, that (as the term "segmentation" indicates) in these cases the goal is to cover the entire image by planes (or other parametric surfaces). When processing elevation maps of natural scenes where irregular terrain occurs, this would typically result in a large number of small planar regions, most of which probably would be irrelevant for our purposes. The more traditional clustering algorithms tend to merge planes with similar orientation or spherical regions with large diameter, which also is not desired here. Furthermore, an important issue in RIS is that of adequately splitting contiguous regions into well-distributed parametric surfaces, e.g. a cylindric shape into planes or a landscape into cubics. "Well-distributed" typically means segments of equal size, or adjacency constraints like continuity of the first k derivatives.

A typical approach to these issues is given by Biswas et al. (1995). Their RIS algorithm is based on quantization of surface normals to some predefined prototype directions. Striking advantages of their approach are its computational simplicity and its parallel structure which yields constant time complexity (with respect to the number of input points) on a SIMD array computer. The use of prototype directions, however, may cause distinct planes to appear as one. Furthermore, spherical surfaces with large diameter will be approximated by planar segments, which is intended by Biswas et al. but inappropriate in our problem.

Jones and Illingworth (1994) employ a three-stage method. First, biquadrics are robustly fit to spatial sectors of depth data. Second, groups of neighboring sectors are formed whose fits yield similar values for properties such as curvature and surface normal direction. Third, these groups are then classified as planar, cylindrical or neither, and the appropriate model is fitted. This method involves a difficult tradeoff between sector size versus reliability of group membership and does not consider the region membership of single pixels, which limits accuracy near 3D edges.

A completely different approach is taken by Hu and Mehrotra (1992). They first perform a segmentation of the range image based on discontinuities in depth ("step edges") and its first spatial derivative ("roof edges"). Surfaces are robustly fitted to the resulting regions. A region merging step concludes the procedure. This method relies heavily on the existence of 3D edges and their localization in a noisy range image. Neither can be assumed in our problem.

All of the above algorithms can be classified as patch fit/merge techniques. A remarkable representative of the parameter clustering methods is the approach of Krishnapuram and Freg (1992). They find an optimum number of parametric clusters in the data, given that an upper bound on the number of clusters present is known, effectively balancing clutter and accuracy. Our problem is different in that the goodness-of-fit of the planes to be found is more likely to be known in advance than the number of them, due to known or assumed sensor and domain characteristics. The objective is more one of focus of attention on a subset of data that must be extracted. Regions that do not possess a sufficiently good fit are to be discarded, not smoothed.

For a comparison of four other state-of-the-art RIS algorithms, refer to Hoover et al. (1996). One other—rather unique—segmentation algorithm should be mentioned (Yu et al. 1994) because it does not fit into either of the previously mentioned categories of approaches. They propose an iterative robust fitting method that randomly samples points in the current *processing region* and solves plane equations for k samples of three points. They present a *compressed histogram* method to select the plane parameters which yield a best *residual consensus.* Initially, the current processing region consists of the entire image. A plane is robustly fitted (using the sampling method) which yields the largest planar region in the image. All other image points are regarded as outliers and form processing regions for consecutive iterations. This process is repeated until the processing regions are exhausted. At this stage, the entire image has been segmented into planar regions. According to the authors, the fitting procedure can handle outlier proportions of around 80%.

All of these algorithms either rely on a heuristic selection of a (typically large) number of parameters which coordinate the different stages of an algorithm, but are reportedly difficult to adjust, or on assumptions about the data (such as an expected number of regions). The main contribution of this research is the development of a concrete, straightforward algorithm for finding planar segments in an irregular, noisy environment that avoids both of these drawbacks. Our goals was the use of plane fits under the same fitting criteria throughout the procedure (rather than arbitrary splitting or merging criteria), and to keep the number of parameters low and their effects intuitively understandable. The following exposition explains the algorithm and discusses advantages and drawbacks of our approach.

2 Algorithm

The algorithm consists of the following largely independent steps:

1. Fit elementary planar *patches.*

2. Merge as many patches to form larger *regions* as appropriate.
3. Refine regions pointwise.
4. Split inconsistent regions.
5. Merge as many regions as appropriate.
6. If a given convergence criterion has not been met, go to 3.
7. Determine the boundaries of the regions.

These steps will be described in more detail in the following sections, accompanied by a sequence of illustrations denoted **Running Example**. Whenever a *plane* is fitted to a set of 3D points, this set will be referred to as a *region*, even though the points are not necessarily connected. The term *patch* is used to refer to the initial regions generated in the first step of the algorithm.

2.1 Elementary planar patches

The entire input image is organized into regularly positioned patches of equal size and shape, which may, but need not, overlap. For each of these patches, a plane is fit to the 3D points within that patch (Fig. 3). If the resulting sum of squared residuals exceeds a given threshold q, then the patch is discarded.

Either least squares or least median square fitting (Doyle 1996) can be used. In the latter case, k planes are hypothesized by randomly selecting three points from a patch. The number k of planes tried is computed as a function of the estimated probability p that any given point is an inlier, and a certainty factor $0 < c < 1$ specifying the reliability desired. The best-fitting plane is used to separate outlier points from inlier points,[1] and then a least squares plane is fitted to the inlier points only. Outliers are returned into the pool of free points. This allows them to be assigned to other planar regions. Generally, the presence of noise dictates that a reasonable minimum number of points be contained in a patch.

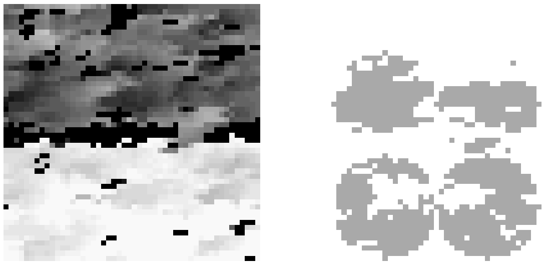

Figure 3: **Running Example:** *Left:* Sample Section of IFSAR data (from Fig. 1). The area measures roughly 25 by 25 meters, sampled by an array of 50 by 50 points. The height range covered is 19 meters. *Right:* Four circular patches were robustly fitted with one pixel overlap in both axis directions. No patch was discarded. Gaps are due to missing data (dropouts) and outliers. The parameters used were $r = s = 10$, $p = 0.8$, $c = 0.9$, $q = 1.5$ (see text).

Important issues are the shape and positioning of patches (Fig. 4). One objective is to minimize the influence of patch orientation on the result (i.e. *patch rotation invariance* is desired). Obviously, circular patches achieve total patch rotation invariance, whereas square patches introduce considerable bias toward straight contours parallel to their edges. The positioning of patches also has some influence on the result. The closer that patches are positioned, or the more they overlap, the better the *patch translation invariance* achieved. Throughout this paper, circular patches with varying degrees of overlap are considered. These issues are strongly related to the merging of neighboring regions which will be discussed in the following section.

2.2 Region merging

Whenever two regions are adjacent (or, at the first iteration, overlap), a merge of these regions is attempted. The objective here is to merge regions iteratively such that as few regions are generated as possible (Fig. 5). This is a very hard optimization task, since every possible combination of pairs of adjacent regions has to be tried in all possible orderings.

Instead, a non-optimal scheme is employed, where a priority queue of potential merges is maintained.

[1] A point is an outlier if its squared residual exceeds the median squared residual of its region by a constant factor; see Doyle (1996) for details.

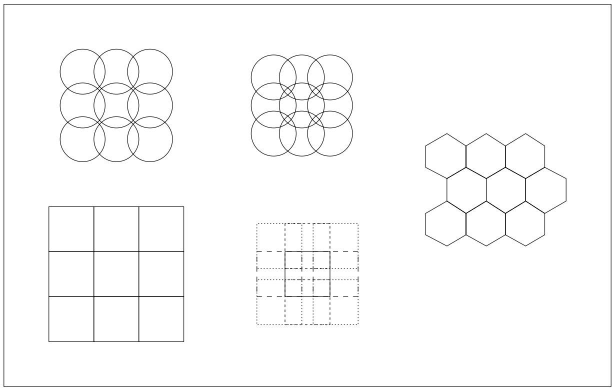

Figure 4: Patch shape and positioning examples: Circular, square and hexagonal shapes; positioning with no overlap (square, hexagonal), 4-neighbor (circular) and 8-neighbor (circular, square) overlap.

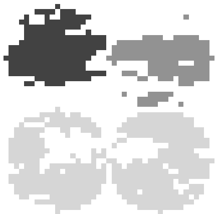

Figure 5: **Running Example:** After the first merging step. Only the bottom two patches were merged, resulting in three planes. (Here and in all following example illustrations, gray levels were chosen to illustrate membership of points to the resulting planes, and are not representative of range, elevation, orientation etc. in any way.)

Suggested merges are taken off the top. One merge is tried at a time by fitting a least-squares plane to the union of the inlier points of the two regions involved. If the resulting mean squared residual does not exceed the threshold q, the new region replaces the two old ones. This is in fact the same criterion as used in the patch fitting step. The least-squares fitting relies on the (reasonable) assumption that the parameters of the merged plane are very similar to those of both the merged planes. In some rare instances, the assumption of similar plane parameters is violated, causing a (typically very small) number of points with large residuals to occur (see Sec. 3.1).

Since the plane parameters of the two merged planes have changed, the merge feasibilities of all remaining potential merges with this new region should be recomputed, thus updating the priority queue. However, since only planes with similar orientations are merged, in general the relative surface orientation is not expected to change much. It appeared a better strategy to avoid the extra cost of updating the priority queue, as suggested by experiments that showed no apparent differences.

The priority queue is initialized before the merging starts. Potential merges must be ranked according to some estimated feasibility f. Intuitively, when two regions lie in nearly parallel planes, f should be inverse proportional to the orthogonal distance between them. On the other hand, if the planes are tilted, there is no well-defined notion of this distance, and f should be inverse proportional to the angle between the planes. The following heuristic equation reflects these intuitions:

$$f = \cos\alpha - \frac{\max\{d_1, d_2\}}{\delta} e^{-\cos\alpha}$$

where α is the angle between the normals of the two planes, d_i is the perpendicular distance of the centroid of region i from the other plane, and δ is a weighting constant. The resulting merge quality is high if both angle and distance are small, and it diminishes as either angle or distance grow. The smaller the angle between the planes, the more important the distance becomes. The value of δ is not critical; the square root of the squared residual threshold q was used here.

Obviously this equation is only a heuristic. A better solution would perform a temporary merge and use the resulting sum of squared errors as the priority. On the other hand, this would slow down the merging process dramatically, since potential fits would be tried in all cases before a single merge takes place. Clearly, the nature of this heuristic function has a significant influence of the final result. It is easy to construct examples where a bad ordering of the merges yields unwanted results (Fig. 6). On the other hand, this rarely happens in practice, since the heuristic works pretty well, as illustrated in Fig. 7. This typical example (from the Lockheed-Martin stereo data, see Fig. 14) illustrates the fact that promising merges are put first in the priority queue: The first 280 merges all succeed. Then, the portion of failed merges increases until most merges fail. At the end, again a small number of merges succeeds. This is a systematic error that needs further investigaion. It may have to do with the aforementioned unadjusted merge ranks and with the heuristic nature of the ranking formula.

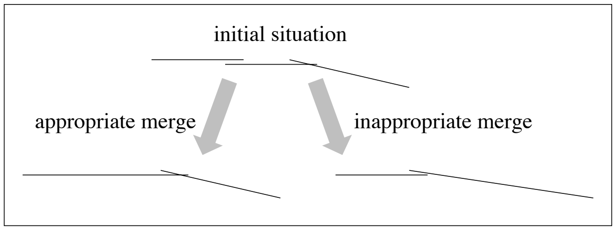

Figure 6: An example of an inappropriate merge, yielding either a good or an unwanted result. The lines represent planar regions, viewed from the side.

Refitting planes to the set of points of the regions that are to be merged is the predominant time factor during the region merging process, especially if very large regions are involved. To avoid unnecessary plane fits, a screening scheme was introduced. Key is a measure m that estimates the confidence that two given planes can be merged in terms of the residual error of the inlier points of the smaller plane with respect to the larger region's plane equation. If this confidence m is very high, then the two regions are merged *without refitting*, assuming that the plane equations do not change much. (Planes of altered regions are refitted periodically.) If m is very low, then the merge is *not even attempted*. In case of medium values of m the merge is attempted as described above, and retained only if the resulting plane satisfies the fit criterion q. Since small regions tend to involve higher uncertainties due to noise, this screening scheme is only applied to sufficiently large regions.

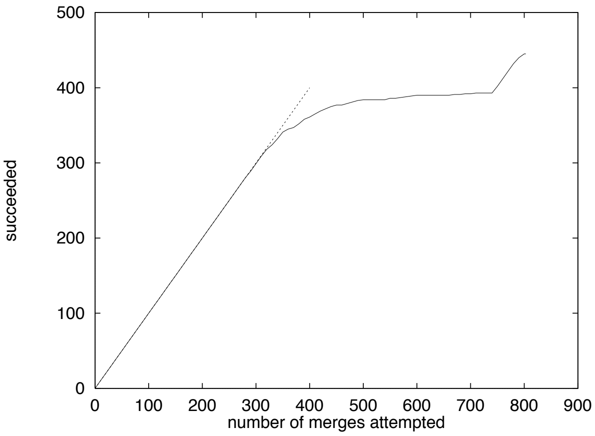

Figure 7: Evaluation of the feasibility measure f for prioritizing potential region merges: Plot of a typical series of patch merges, total attempted versus successful merges. The dashed line indicates a slope of one for comparison.

Just as the shape of a patch influences the patch rotation invariance, the neighborhood relationships defining the regions which can be merged will affect the *merge rotation invariance* (see Fig. 4). This is easiest to see when elementary patch merges are considered: In the simplest case, merges are performed only in 4-neighborhoods (assuming grid-like patch positions). This favors regions with boundaries parallel to the patch grid axes. Diagonal boundaries will be approximated as staircase-like shapes.

For maximum merge rotation invariance, 8-neighborhoods (or even higher) should be used. Since it only makes sense to merge patches that share a piece of boundary or overlap, this question is related to the topic of patch shape and patch overlap. Obviously, it makes no sense to use 8-neighborhood merges with non-overlapping patches. Therefore overlapping regions play a key role in region merging.

An unsolved problem with overlapping regions is, when a merge fails, how to deal with those points that belong to both regions. For simplicity, the algorithm presented here removes these points from both regions and relies on the region refinement step (described in the next section) to fill any gaps. With 8-neighborhoods, in certain situations a region may be split into disconnected subregions, as illustrated in Fig. 8.

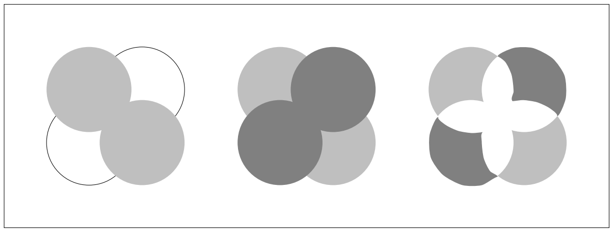

Figure 8: The problem of crossing regions. Suppose two diagonally overlapping patches have been merged, as shown on the left. Suppose now that the other two regions are merged next (middle). Then, the two resulting regions are tried, but this merge fails. Their intersection is removed from both regions, and we are left with two point sets each of which has been cut into two regions (right).

Hexagonal patches seem to constitute a reasonable compromise for non-overlappinng patches. Their unique 6-neighborhood also achieves some sense of rotation invariance. However, its characteristics and problems are very different from grid-like patch arrangements, and will not be considered in this discussion.

2.3 Region refinement

The region refinement step simultaneously serves three purposes, namely

- adding free points adjacent to a region if they are perpendicularly close enough to the planar surface,
- removing points currently belonging to a region if they are too far away from the surface (this is primarily useful if non-robust fitting was employed), and
- moving points at the juncture of two regions from one to the other if they are more properly part of the other.

There are two possible reasons for taking a point u out of a planar region A: One is that u is further away from the plane of A than some threshold, and the other is that there is a plane B that u is closer to than A. (To discourage oscillations, a hysteresis mechanism was implemented which is not discussed here.) Initially, all points are unflagged. For every unflagged point u, the following procedure is performed (Fig. 9):

```
Let $A$ be the current region to which $u$ belongs,
    or NIL if $u$ is a free point.
Let $B$ be the region $u$ is determined to fit best to,
    or NIL if none.
If $B$ is not equal to $A$, then:
    If $A$ is not NIL:
        If there is no reason to take $u$ out of $A$:
            continue with next point $u$.
        Take $u$ out of $A$.
    If $B$ is not NIL:
        Insert $u$ into $B$ and flag $u$.
```

The convergence criterion mentioned at the beginning of Sec. 2 is defined as a threshold to the number of points changed (i.e. added, removed, or moved) during one iteration of the above procedure. If fewer points than this threshold were changed during a given iteration, convergence is assumed, and the algorithm terminates.

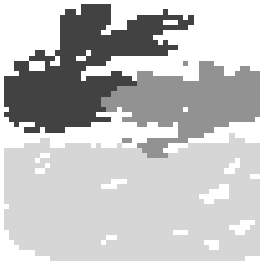

Figure 9: **Running Example:** After the first region refinement step. Here, all three regions have grown considerably and are mostly contiguous.

2.4 Region splitting

A region needs to be split if it is crossed by a contiguous chain of points that do not belong to it. Such situations can occur for three reasons: These points are outliers, or they were created during the region refinement step, or they were removed from the region following a failed attempt to merge it with a crossing region (see Sec. 2.2). Here, a 4-Connected-Components operation (modified from Ballard and Brown 1982) is performed. Besides finding connected regions, it determines adjacency of different regions. After splitting regions into connected components, a new priority queue of potential merges is constructed from these adjacencies. After all appropriate merges have been performed, all regions smaller (i.e. containing fewer inlier points) than a given threshold are discarded. Then, the next refinement step begins (Fig. 10).

2.5 Boundaries

All previous steps are performed without maintaining explicit lists of boundary points. The construction of these lists is the final step of this algorithm. All boundaries are determined, including those of interior subregions not belonging to the region in question ("holes"). The lists are constructed such that points to the left of the boundary points belong the region, and points to the right do not. In other words, the hull is traversed counterclockwise, and holes are traversed clockwise. (This interpretation assumes that the (i, j) coordinate system of the input point grid is right-handed.) Boundary points play a crucial role in visualization as well as many applications, as shown in the following section.

3 Sample results on real data

In the following sections, the performance of this algorithm is illustrated using typical data samples of different types. Before these tests were performed, all screening parameters were adjusted such that they achieve their purpose but do not unduly influence the

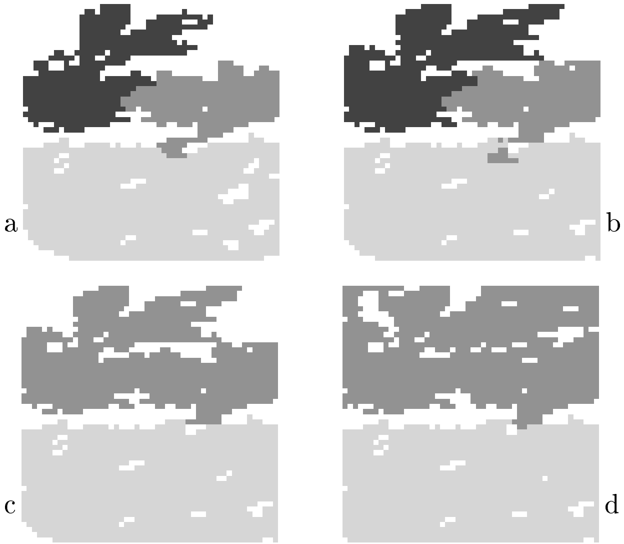

Figure 10: **Running Example:** After region refinement, the first splitting step split two small "islands" off the medium-gray region (Fig. 9). They were returned into the pool of free points because they were smaller than the minimum region size (a). Then, the second iteration of the refinement/split/merge sequence starts: Refinement grows all regions (b), and some of the points formerly belonging to the medium-gray region have migrated to the light gray region (note the dark "island" in the light gray region). The next split/merge sequence merges the two upper regions and merges the "island" to the surrounding region (c). —This sequence is iterated until convergence (in this case, five iterations) and results in the configuration (d). The height of the dark fitted region ranges from 1646.6 to 1648.1 meters, and the light region 1652.5 to 1654.6 meters.

result. These parameters are intended to be fixed across data sets. In all of the following examples, overlapping circular patches are used in the initial patch fitting step. The only user-specified parameters are patch radius r and offset s (which also affects whether or not diagonal patch merges are considered), and the squared residual threshold q. The parameters r and s should generally be set as large as possible, but small enough to catch the smallest desired planar region. For robust fitting, p and c (which map onto only one parameter, as explained above in Sec. 2.1) can be adjusted to reflect the given noise characteristics. Since one of the design goals of this algorithm was its independence from heuristic parameters that need to be fine-tuned for specific input data, it must be emphasized that no extensive systematic or empirical parameter tuning was performed to improve the results shown in the following sections. In all cases, unless otherwise noted, the parameters were manually selected in a straightforward fashion.

3.1 IFSAR

As mentioned in the introduction, IFSAR data such as the Kirtland data set poses a challenge to any plane fitting and segmentation algorithm due to its speckled non-Gaussian noise and dropouts. Both noise and dropouts occur in chunks of several correlated data points. In particular, dropouts often form relatively large contiguous regions. Some tests were run on IFSAR data from Sandia/Kirtland AFB (see Fig. 1). This site is characterized by many planar faces parallel to the ground.

The result is presented in Fig. 11 (with the same parameter settings as in the Running Example above). Note the many holes caused by outliers and dropouts. Most of the ground plane and the “E”-shaped building complex in the middle are represented by large connected regions. Most of the semi-cylindrical Quonset huts (white buildings at the bottom and right of Fig. 11c) are covered only in parts and mainly show up as gaps in the ground plane. This is not surprising since the corresponding IFSAR data mainly consist of dropouts.

Figure 12 shows a histogram of the residuals of the points belonging to the fitted planes. Except for very few points with large residuals (up to almost 11 meters), the residuals are pretty low (less than 1 meter). A total of 1164 points (about 0.6 percent of all points) have a residual larger than 3 standard deviations, which is about 2 meters. These residuals are artifacts created by the least-squares fitting in the merging step in conjunction with the non-optimal merging queue (cf. Sec. 2.2). Ideally, they should be removed from the regions. However, in rare cases this could create disconnected subregions, which in turn (during refitting) might produce outliers. This problem is not easily solved.

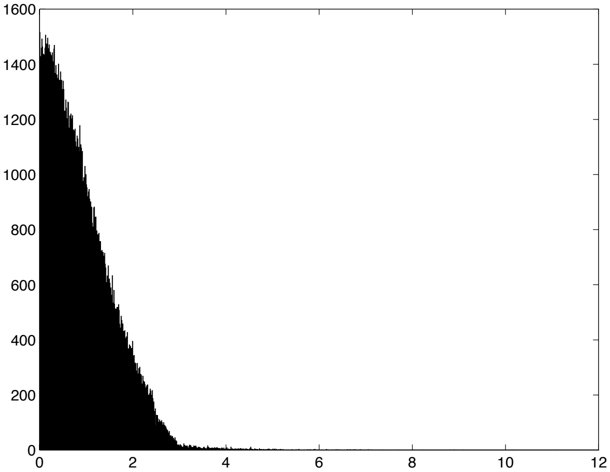

Figure 12: Histogram of the residuals (in meters) of the 185318 points in Fig. 11 that belong to any fitted plane. The residuals range from 0 to 11.1 meters, with mean 0.9 and standard deviation 0.7 meters.

Interestingly, it is relatively easy to separate the ground planes from other planes. This feature can serve as a simple focus-of-attention mechanism for building detection. For example, the user could select a region that is believed to belong to the ground plane. This region is then marked as a ground region, along with all regions “similar” to it. The similarity measure here is exactly the same parameter q as used during region merging (cf. Sec. 2.2), the only difference being that here pairs of regions need not be connected. What remains are the regions that do not belong to the ground plane, i.e. that are likely to

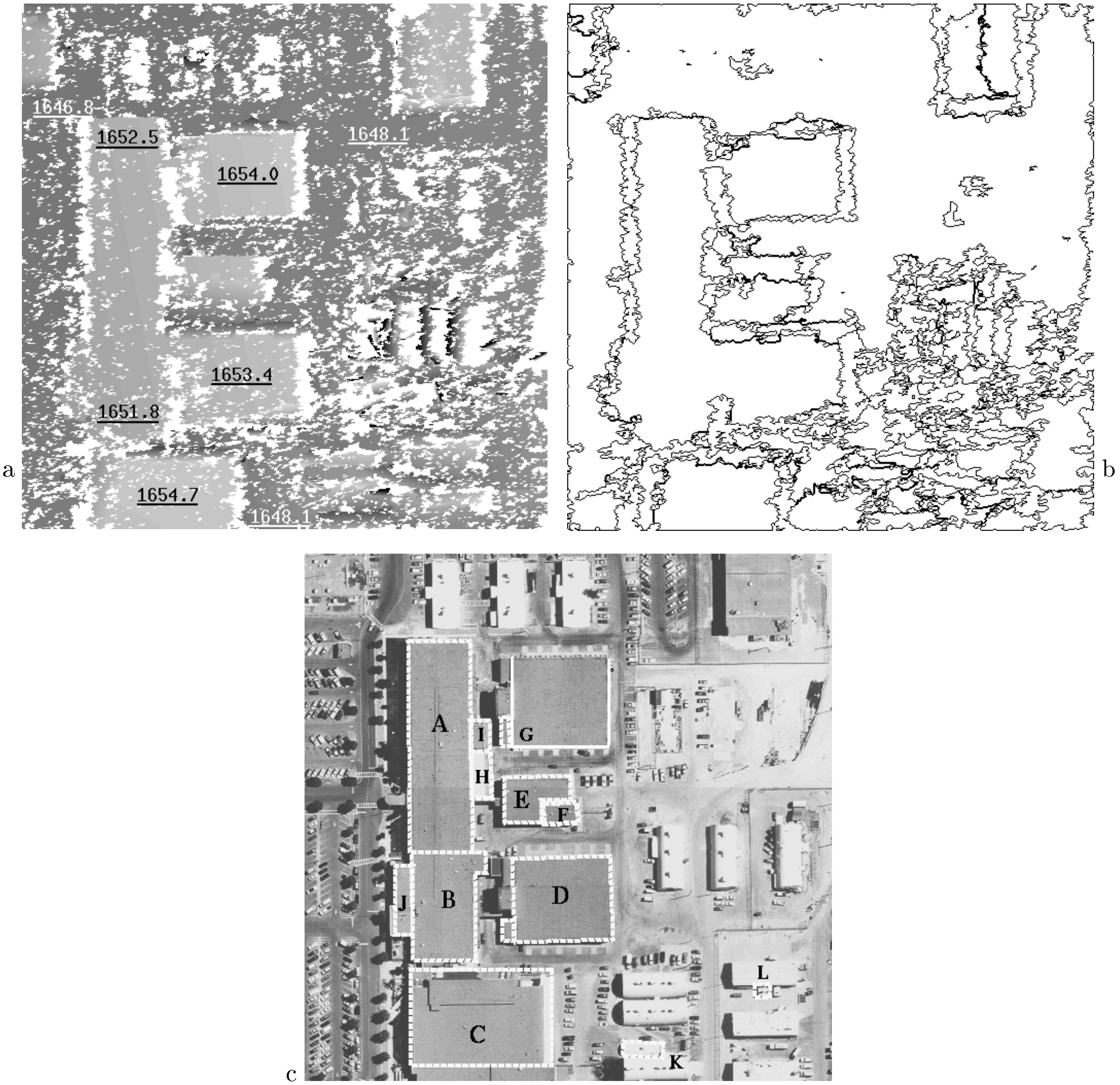

Figure 11: Fitted regions (nonoverlapping patches, $r = s = 10$, with horizontal/vertical adjacencies): (a) dark to light gray levels encoding increasing elevations (white areas are not covered by any fitted plane). Some heights (in meters) are given; (b) region outlines. Note that no geometric constraints are used to straighten lines etc. The image size is 512 by 512 pixels; (c) an optical image of the same site for illustration.

belong to buildings. Some sample results of a ground plane removal from Fig. 11 are shown in Fig. 13. The ground region (the largest region in Fig. 11b) was selected as the ground prototype. The higher the value of q that is used, the less stringent the similarity measure for plane comparison is, and consequently, the more planes are taken as belonging to the ground and discarded.

3.2 UMass DEM

The input 3D data considered in this section were computed by the UMass terrain reconstruction system (see introduction). In the context of building detection or reconstruction, a possible application of the plane fitter is to focus attention on restricted regions of such DEM's. This can be desirable in situations where image data provides insufficient cues or a lot of clutter. In this case, one wants to set the fit threshold q to a very small value, resulting in a sparse coverage of the image by planar regions. To demonstrate this, an image of a Lockheed-Martin building located in a rural area was selected (Fig. 14). Patch radius and offset were $r = s = 5$ pixels which yields horizontally, vertically and diagonally overlapping patches. Because the data contain little noise and we are interested in finding buildings, least squares fitting was used, and the fit threshold was set to a very strict $q = 0.01\text{m}^2$.

Indeed, the fitted regions cover only about one quarter of the image, and the building is clearly represented (in the center of the image). Note that the flat pavement surrounding the building is not covered by any fitted planes. This is because the lack of texture in these locations produces increased noise in the elevation data. — From this point onward, a terrain reconstruction algorithm (for example) need process only the regions covered by fitted planes. Furthermore, part of the roof of the Lockheed-Martin building is already described parametrically as 3D surfaces in the appropriate regions. This demonstrates the suitability of this algorithm as a planar object detector or focus-of-attention mechanism.

Next, the plane fitter was run on a peaked-roof building from ISPRS data. The reconstructed digital elevation map was generated by the UMass terrain reconstruction system (see Fig. 2). The plane finder was run on this tile with several different parameter configurations. The patch offset in the axis directions was set equal to the patch radius such that horizontal, vertical, and diagonal neighbors overlapped by half a patch width. Patches were fitted using the least-squares algorithm, since these data contain little random noise.

Figure 15 shows the boundaries of the fitted regions generated for two different mean squared error thresholds q and two different patch radii r, overlaid on a slightly brightened elevation image. In the illustration, the boundary overlays are one pixel wide. Where these lines appear bold, two such boundaries of neighboring regions meet. Note that many regions contain holes, and that not quite the entire image is covered by fitted regions. It is apparent how increasing error thresholds yield larger regions. Note how nicely the roof slopes are fitted at $q = 0.5$, but how much detail is lost compared to $q = 0.1$. Starting off with smaller patches yields even more detail. The slopes of the roof are consistently found to be approx. -25 and 55 degrees. This asymmetry is inherent in the elevation data (cf. Fig. 2).

A completely different example was taken from a Fort Hood data set (UMass tile 23; Fig. 16). This sample was a lot harder to deal with because the elevation data is much noisier here than in the ISPRS case, and there are many larger bumps and dents in the elevation data. Because of the noise, least median square fitting was used ($p = 0.8$, $c = 0.9$). The fitting threshold was set to $q = 0.5$. The patch diameter was $r = 8$ pixels, and the patches did not overlap since it did not appear necessary because the patches are relatively small compared to the expected size of the recovered planar regions.

A different section of the Ft. Hood data is shown in Fig. 17. Here, the elevation data contains a lot more noise than in the previous case due to distant oblique views and poor contrast in the stereo images (Fig. 17a shows one of them). It was processed using the same settings as above, but using slightly larger patches ($r = s = 10$). All buildings shown in this scene have peaked roofs at shallow angles. Note that many of them are fitted in spite of the noisy data. The resid-

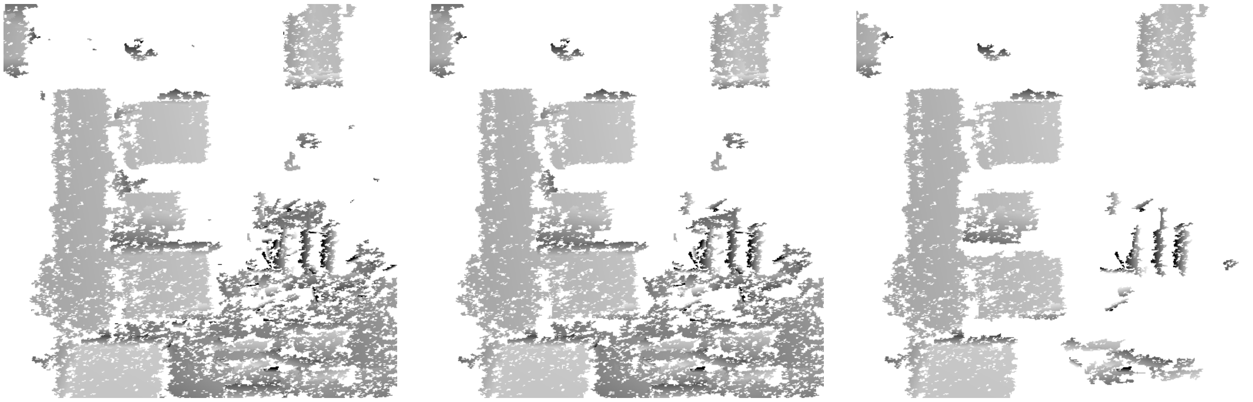

Figure 13: Same data as Fig. 11, but with ground regions removed with $q = 1$, 1.5, 2 (left to right). Lighter gray levels indicate higher elevation; white areas are not covered by any region. Note that $q = 1.5$ is the same value as used for the region generation (see text).

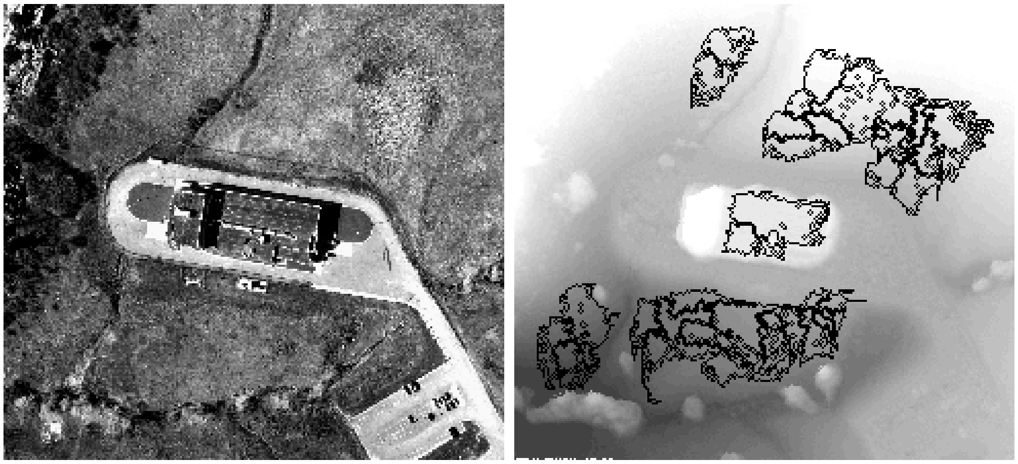

Figure 14: Lockheed-Martin building. Aerial image and (registered) DEM overlaid by plane finder result (244 by 222 pixels).

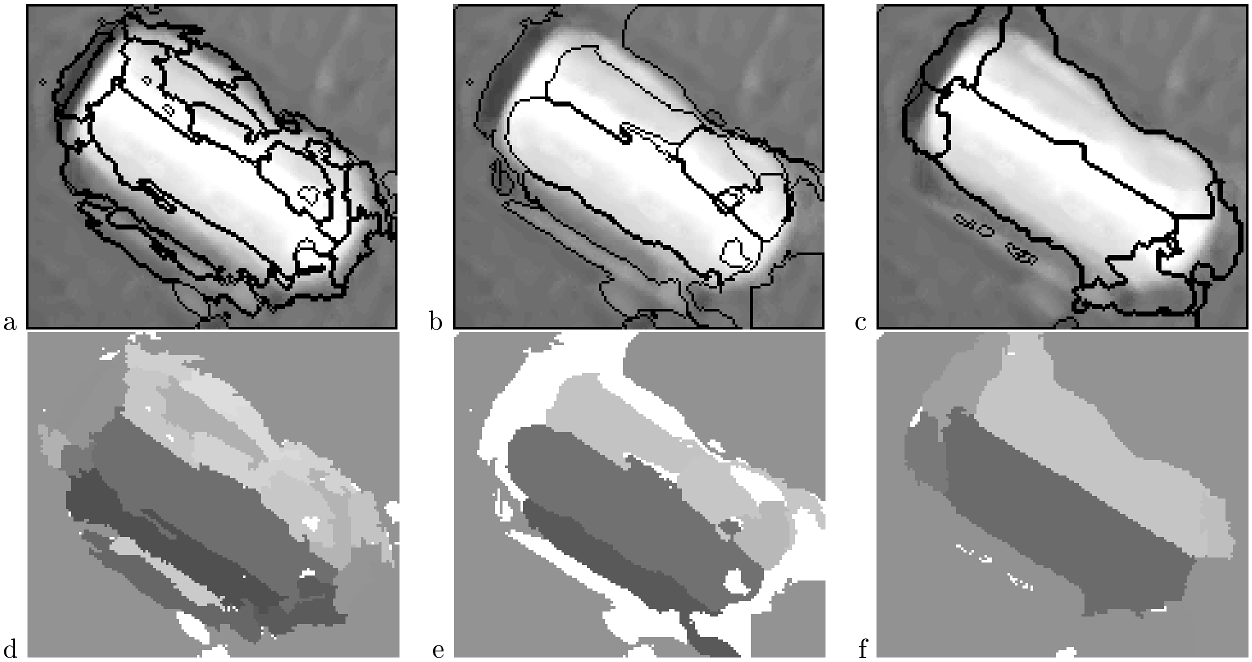

Figure 15: ISPRS stereo data: (a-c) Region boundaries overlaid on a slightly brightened elevation image (145 by 127 pixels). Least squares patches; (a) $r = 5$, $q = 0.1$, (b) $r = 10$, $q = 0.1$, (c) $r = 10$, $q = 0.5$. (d-f) Visualization of plane orientations: Gray level encodes the angular deviation of a fitted plane from an imaginary vertical plane through the ridge of the house. Angles are mapped into the range of -90 (dark gray) to 90 degrees (light gray). White areas are not covered by any fitted plane.

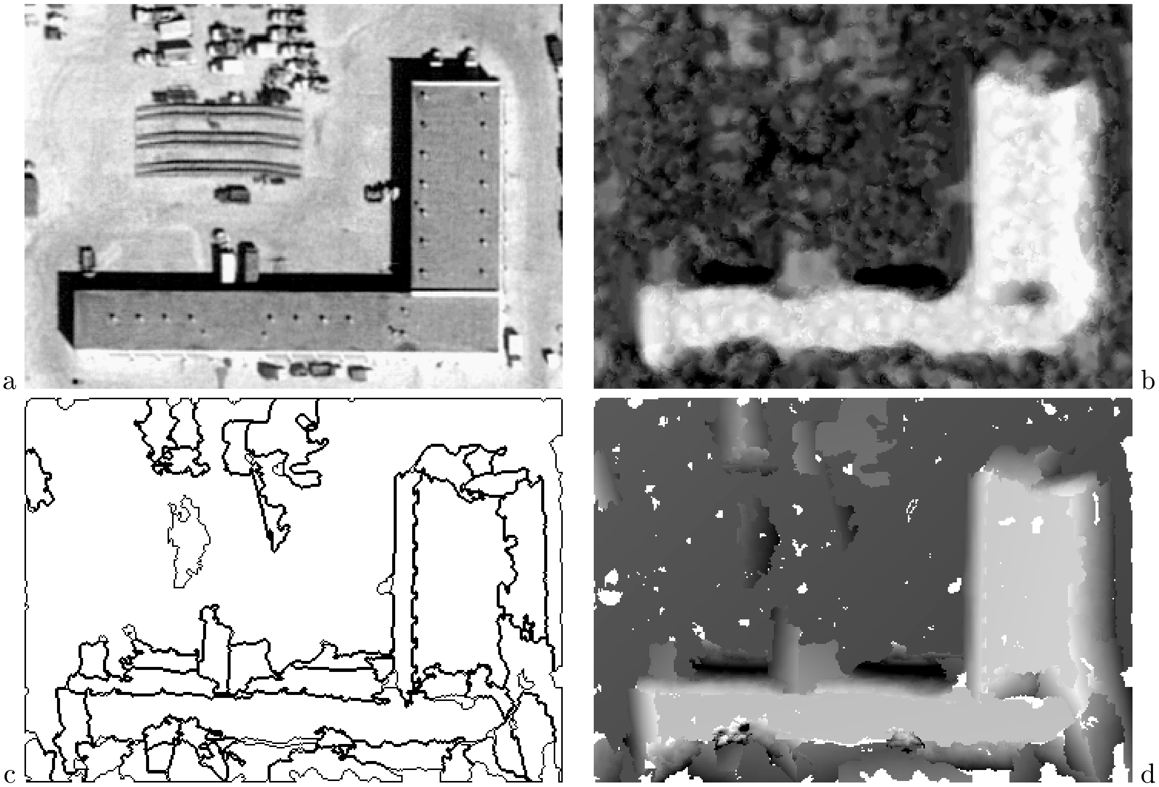

Figure 16: Fort Hood flat-roofed building (350 by 250 pixels): (a) visible image, (b) stereo DEM, (c) region outlines (holes are not shown), (d) fitted planes (white areas are outliers or other points not belonging to any fitted plane).

ual statistics are shown in Fig. 18. They are very similar to those shown in Fig. 12: 1336 point residuals (less than 0.6 percent) are larger than 3 standard deviations (1.2m).

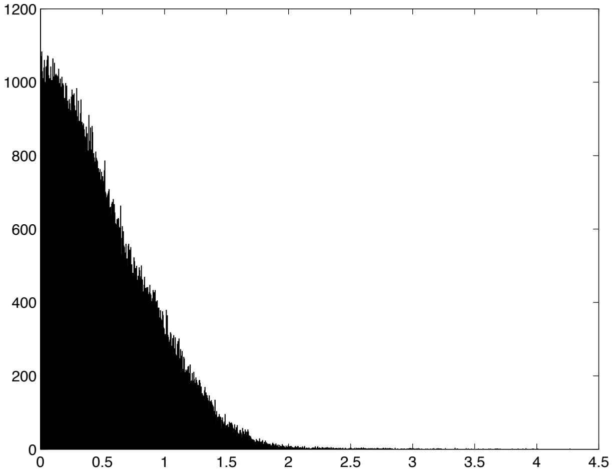

Figure 18: Histogram of the residuals of the 194788 points in Fig. 17 that belong to any fitted plane. The residuals range from 0 to 4.3 meters, with mean 0.5 and standard deviation 0.4 meters.

3.3 Range Data

Hoover et al. (1996) developed a testing scheme for range image segmentation algorithms, including a number of carefully created 512 by 512 range images of polyhedral objects. These range data show large planar faces that cover most of the image area. They are characterized by relatively crisp edges and little random noise. Depending on the type of the range finder, large shadow areas without depth information may occur. Even though this plane finder was not designed for range images, two different sample images were chosen and segmented using non-overlapping patches of radius $r = 10$ and least squares fitting with a coarse threshold of $q = 5$. The first image (Fig. 19) was generated by an ABW Structured Light Camera, and the second (Fig. 20) by a Perceptron Laser Range Finder. While the results appear satisfactory for our purposes, they illustrate some disadvantages of our approach if applied to segmentation. In both cases, almost all faces are covered, and many edges are straight and well-defined. These results are remarkably good, given that no attempt was made to detect edges in the 3D data. On the other hand, some edges are very inaccurate. True RIS algorithms perform better because they typically make explicit use of edges, which our plane finder intentionally avoids. Because of this intrinsic difference between our algorithm and typical RIS algorithms, no attempt was made to formally assess the results using the framework and tools provided by Hoover et al.

4 Interaction with higher-level procedures

As mentioned in the introduction, the plane finder can be used to supply additional information to terrain reconstruction systems. Here is one example: *Ascender* (Collins et al. 1996) is a site reconstruction system that detects straight lines in aerial (optical) images, groups them into polygons and generates building hypotheses. In the visible image data shown in Fig. 11c, Ascender failed to detect the upper rooftop edge of the building labeled “G”, because loading docks disturb the straight appearance of this edge. To verify the building hypothesis, the location where a “U”-shape boundary is found is cropped out of the data, and the plane finder is run. A subimage containing the result (Fig. 21) shows that indeed a planar surface was found at the appropriate location, with three of its edges corresponding to the roof edges hypothesized by Ascender. The fourth edge of the planar region provides the missing rooftop edge. Hence, the existence of a building is confirmed.

While the plane finder operates well on 3D point data without additional information, it is easy to make use of additional knowledge if available. For example, if optical imagery is available, lines (grayscale edges) can be extracted from them and grouped to form open or closed polygons as above. In many applications, it is reasonable to assume that 3D edges coincide with grayscale edges in the visible image. An easy way to communicate this cue to the plane finder is to remove all points from the 3D data that coincide with lines in the optical image. This prevents the

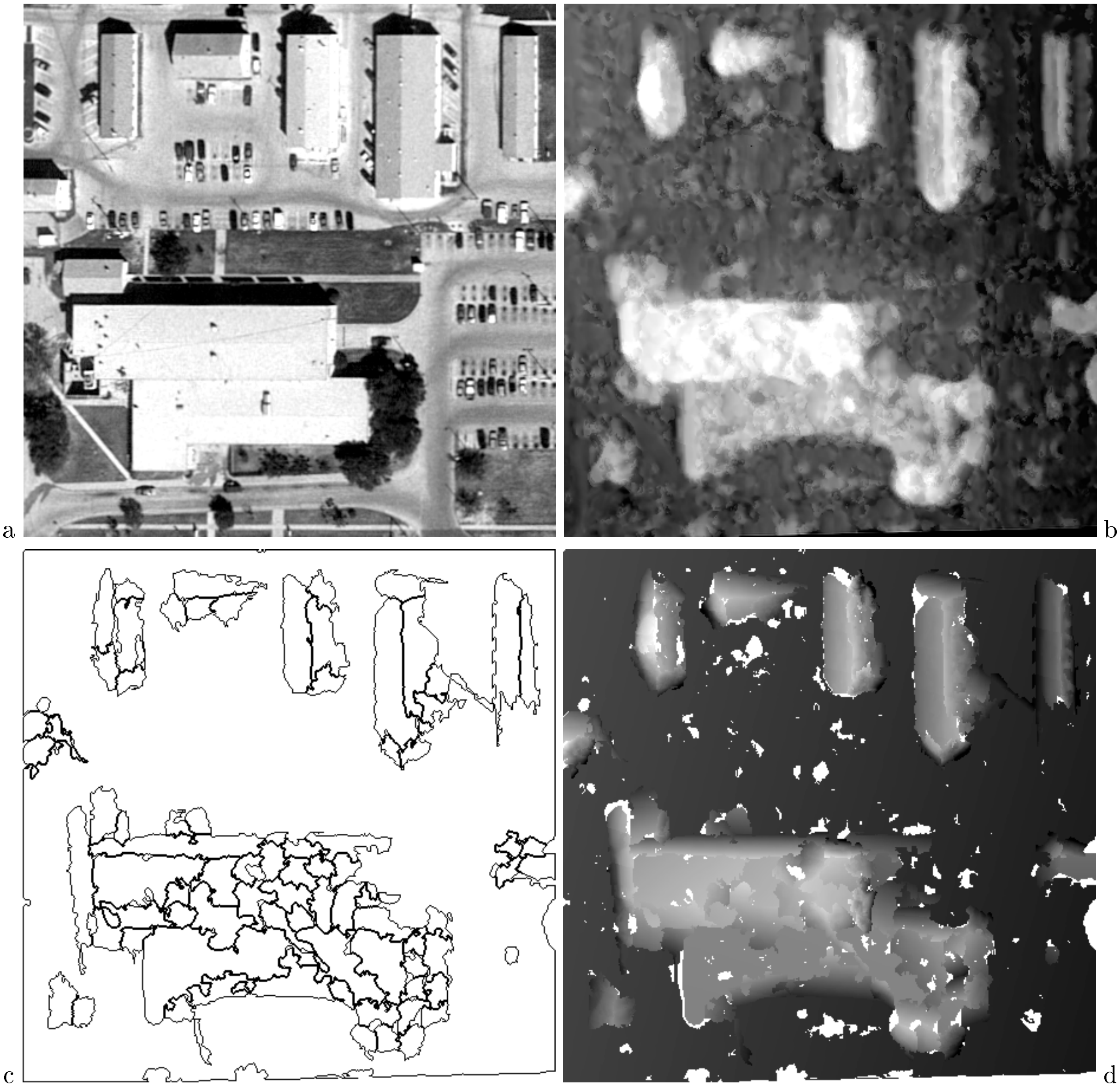

Figure 17: Fort Hood peaked-roof building (451 by 451 pixels): (a) visible image, (b) stereo DEM, (c) region outlines (holes are not shown), (d) fitted planes.

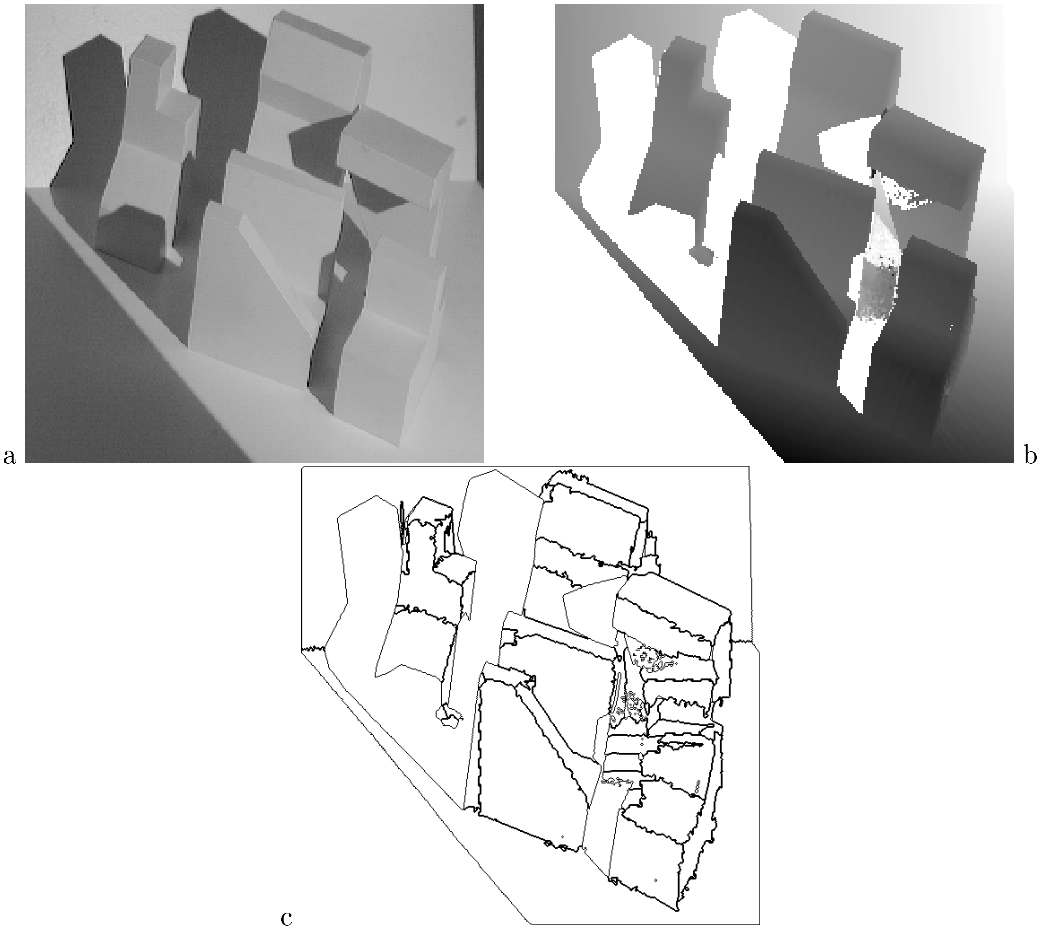

Figure 19: (a) Intensity image, (b) ABW range image (in white areas, no range information is available), (c) planes recovered from the range image.

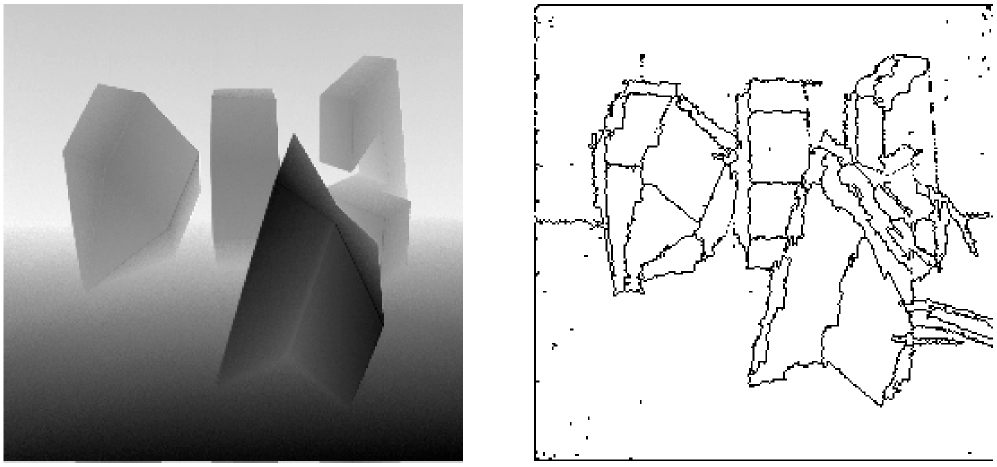

Figure 20: Left: Perceptron range image; right: plane finder result.

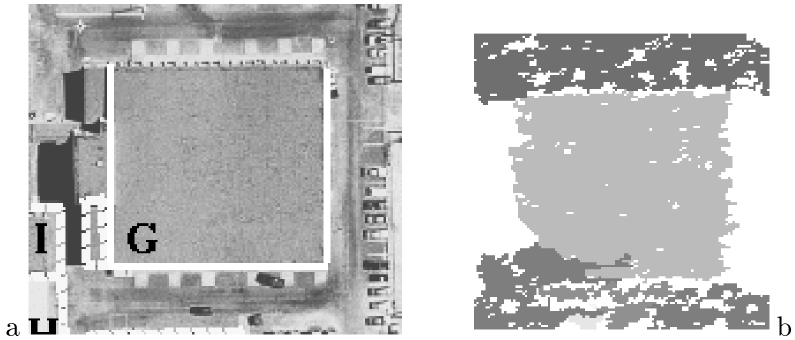

Figure 21: (a) Subimage of Fig. 11c (Kirtland AFB site). White solid lines indicate roof edges of building G which were found by the Ascender system. The upper side was not found because its appearance is disrupted by a series of loading docks. The other three sides were found and grouped as a U-shape, providing a building hypothesis that can be verified using the IFSAR data in this area. (b) Verification of the hypothesized building "G" and detection of the missing upper side (the upper horizontal edge of the square region; see text). The white areas are gaps, outliers or dropouts. Gray levels encode elevation.

plane finder from generating regions that cross these lines, thus improving the accuracy of the planes near edges.

Figure 22 shows an example result of this procedure applied to the ISPRS data. First, straight lines were extracted from one of the two optical stereo images using Burns' (1986) algorithm. Then, these lines were removed from the elevation map, and the plane finder was run with the same parameters as used in Fig. 15c. The improvement is striking: Without the line extraction, the plane finder had difficulties precisely locating the region boundaries because of the absence of sharp edges in the 3D data (cf. Fig. 15). After removing the lines, the accuracy of the outlines of both halves of the roof is greatly increased.

Note that this procedure is robust with respect to spurious edges: If a large number of edges is extracted that have no corresponding edge in 3D, all that happens is that neighboring regions are not merged even though they lie in (nearly) identical planes. To account for this, one can easily add a postprocessing stage that inserts the missing points into the data, adds them to the region they fit best to (if any), and then tries to merge neighboring regions.

5 Limitations, drawbacks, and improvements

The strength of the presented algorithm is also its greatest weakness: As mentioned above, a strong design principle was the use of plane fits instead of heuristic merging criteria wherever possible. This accounts for the sometimes poor representation of edges, even if they are very clear in the input data. More care should be taken to accurately locate edges where they are expected; one possible approach was presented in the previous section.

Another drawback is the computational complexity, especially if robust fits are performed. For small images (up to, say, 100 by 100 pixels, depending on CPU speed) performance times are not critical. An image is processed within a few seconds. However, for larger images (say, 512 by 512 pixels), the running time increases considerably. Especially if many patches are generated and many merges succeed (such that regions grow large), the region merging process easily takes more than thirty minutes on a fast SGI (processor speed 200MHz). This step is by far the most time-consuming one. It should be noted, however, that this algorithm can be highly parallelized on many levels. If the plane finder is used as a subsystem in top-down processing (e.g. as illustrated in the preceding section), then the procssing regions will often be small, and thus running time becomes a minor issue.

If least-median-square plane fitting is used, the patch fitting step takes many times longer (up to some minutes) than in the least squares case. This factor depends on the p and c parameters (cf. Sec. 2.1) and will typically lie between about 3 and 10.

The iterative nature of the refine-split-merge procedure does not appear to be a significant time factor, since it usually converges after a small number of iterations. The main time consumer, the region merging step, becomes less significant with growing number of iterations for two reasons: The number of potential merges decreases, and more and more potential merges are rejected by screening (cf. Sec. 2.2) such that the costly fitting step is avoided.

High memory requirements are characteristic of most image processing procedures. This is very significant here because during the iterative region merging and refinement steps, non-local computations on large regions occur which may cause a machine with limited memory to swap extensively.

The merge quality ranking criterion (Sec. 2.2) is somewhat arbitrary. However, it is not clear how to specify a consistent one that relates to the plane fit threshold q in a well-defined way. This problem is shared by all comparable region merging approaches to range image segmentation. Note, however, that in our algorithm this criterion affects only the *ordering* of the merges rather than the merges themselves. This constitutes a clear improvement over other region merging approaches.

The larger the intersection of two overlapping patches, the more points will be deleted from these two regions if they cannot be merged. Thus, gener-

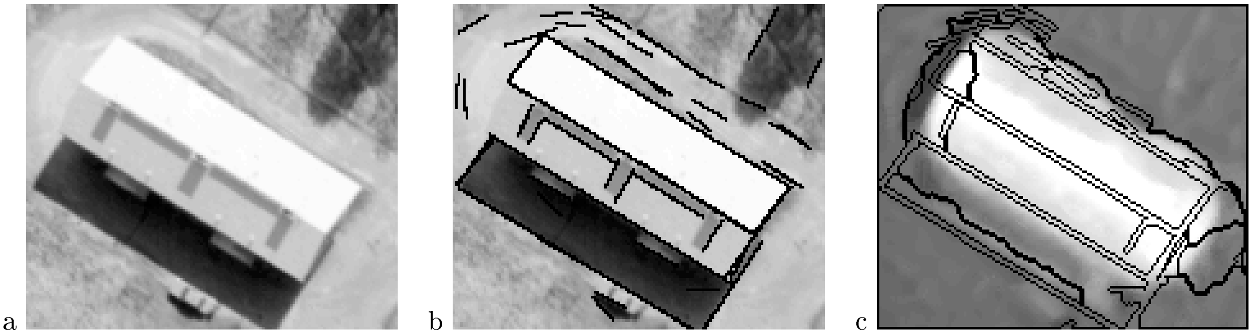

Figure 22: ISPRS peaked roof house (compare with Fig. 15b). (a) Optical image; (b) optical image overlaid with extracted lines; (c) brightened elevation map overlaid with boundaries of fitted regions (holes are not shown).

ally far more points are removed than necessary. A simple solution would be to delete the points from just one region, e.g. the smaller one. Or, points can be kept in the closer region only. More complex approaches may analyze the residual distributions of the points in the overlapping segments, and try to assign the points to the region of best fit. This is basically what is attempted in the region refinement step (Sec. 2.3).

At the end of the iterative refinement/merging procedure, small holes should be included as outliers, rather than tracing their boundary points.

An important improvement would be additional functionality for hypothesizing straight line (or other parametric) boundaries in a bottom-up manner. An iterative procedure similar in spirit to the plane finder could be employed for this purpose.

6 Conclusion

A straightforward algorithm was developed and implemented that finds planar regions in 3D elevation data. It does not make strong assumptions about the nature of the data, and works well on very different and challenging classes, including highly noisy IFSAR and stereo-generated DEM. The desired amount of detail can be scaled by patch radius, patch overlap, and an error threshold. Noise characteristics can be taken into account by selecting an appropriate number of iterations for the robust fitting procedure. The algorithm reflects the desire to keep the results plausible and consistent in that heuristics are mainly used for speedup and are not observed to have much influence on the quality of the results. Wherever possible, the same fitting criteria were applied during all major steps (patch fitting, region merging, and region refinement). The resulting planes seem convincing. In addition to purely data-driven planar region extraction, the algorithm can also operate as a supplementary tool as part of a higher-level site reconstruction procedure by incorporating edge information, assuming the presence of straight line or rectilinear boundaries, or by supplying additional 3D information.

If relatively few and small planar regions are found in a given image, run-time and memory requirements are minor issues. If large regions are found in a large image, then the running time increases quickly. On the other hand, the algorithm can make use of massive parallelism. Further suggestions for improvement were made in the preceding section.

References

Ballard, D. H. and C. M. Brown (1982). *Computer Vision.* Englewood Cliffs: Prentice-Hall.

Biswas, P. K., S. S. Biswas, and B. N. Chatterji (1995). A SIMD algorithm for range image segmentation. *Pattern Recognition 28*(2), 255–267.

Boulanger, P. and G. Godin (1992). Multiresolution segmentation of range images based on Bayesian decision theory. *Proceedings of the SPIE – The International Society for Optical Engineering 1825*, 338–350.

Burns, J. B., A. R. Hanson, and E. M. Riseman (1986). Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*(4), 425–455.

Collins, R. T., A. R. Hanson, E. M. Riseman, C. O. Jaynes, F. Stolle, X. Wang, and Y.-Q. Cheng (1996, Feb.). UMass progress in 3D building model acquisition. In *ARPA Image Understanding Workshop*, Palm Springs, CA.

Doyle, J. K. (1996). Selecting robust plane-fitting algorithms. Computer Science Technical Report, forthcoming, University of Massachusetts at Amherst, Amherst, MA 01003.

Giglio, D. and R. Carlisle (1995). Topographic mapping using interferometric synthetic aperture radar. *Proceedings of the SPIE – The International Society for Optical Engineering 2487*, 381–392.

Hoover, A., J.-B. Gillian, X. Jiang, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. Eggert, A. Fitzgibbon, and R. Fisher (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*(7), 1–17.

Hu, F. and R. Mehrotra (1992). An integrated approach to range image segmentation. In R. Zurawski and T. S. Dillon (Eds.), *Proceedings of the IEEE International Workshop on Emerging Technologies and Factory Automation*, Aldershot, UK, pp. 493–497. CRL Publishing.

Jaynes, C. O., F. R. Stolle, and R. T. Collins (1994). Task driven perceptual organization for extraction of rooftop polygons. In *IEEE Workshop on Applications of Computer Vision*, pp. 152–159.

Jones, G. A. and J. Illingworth (1994). Robust segmentation of curved surfaces from range data. In S. Impedovo (Ed.), *Proceedings of the 7th International Conference on Image Analysis and Processing.*, Singapore, pp. 453–456. World Scientific.

Kaveti, S., E. K. Teoh, and H. Wang (1994). On 3-D image segmentation based on planar and quadric approximations. In *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, Volume 1, New York, pp. 349–353. IEEE: IEEE Press.

Krishnapuram, R. and C.-P. Freg (1992). Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition 25*(4), 385–400.

Schultz, H. (1995). Terrain reconstruction from widely separated images. *Proceedings of the SPIE – The International Society for Optical Engineering 2486*, 113–123.

Yu, X., T. D. Bui, and A. Krzyzak (1994). Robust estimation for range image segmentation and reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*(5), 530–538.