Comparison of an $\ell^1$-regression-based and a RANSAC-based Planar Segmentation Procedure for Urban Terrain Data with Many Outliers

Jian Luo$^a$, Zhibin Deng$^a$, Dimitri Bulatov$^b$, John E. Lavery$^{c,a}$ and Shu-Cherng Fang$^a$

$^a$ Edward P. Fitts Department of Industrial and Systems Engineering
North Carolina State University, Raleigh, NC 27695-7906, USA

$^b$ Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Gutleuthausstr. 1, 76275 Ettlingen, Germany

$^c$ Mathematical Sciences Division and Computing Sciences Division
Army Research Office, Army Research Laboratory
P.O. Box 12211, Research Triangle Park, NC 27709-2211, USA

ABSTRACT

For urban terrain data with many outliers, we compare an $\ell^1$-regression-based and a RANSAC-based planar segmentation procedure. The procedure consists of 1) calculating the normal at each of the points using $\ell^1$ regression or RANSAC, 2) clustering the normals thus generated using DBSCAN or fuzzy $c$-means, 3) within each cluster, identifying segments (roofs, walls, ground) by DBSCAN-based-subclustering of the 3D points that correspond to each cluster of normals and 4) fitting the subclusters by the same method as that used in Step 1 ($\ell^1$ regression or RANSAC). Domain decomposition is used to handle data sets that are too large for processing as a whole. Computational results for a point cloud of a building complex in Bonnland, Germany obtained from a depth map of seven UAV-images are presented. The $\ell^1$-regression-based procedure is slightly over 25% faster than the RANSAC-based procedure and produces better dominant roof segments. However, the roof polygonalizations and cutlines based on these dominant segments are roughly equal in accuracy for the two procedures. For a set of artificial data, $\ell^1$ regression is much more accurate and much faster than RANSAC. We outline the complete building reconstruction procedure into which the $\ell^1$-regression-based and RANSAC-based segmentation procedures will be integrated in the future.

Keywords: DBSCAN, fuzzy $c$-means, $\ell^1$ regression, outlier-rich, planar, RANSAC, segmentation, urban terrain

1. INTRODUCTION

Context-based segmentation of urban terrain has been an issue of intensive scientific research in recent decades because of a wide spectrum of applications including, prominently, urban planning and automatic low-altitude navigation. Point clouds created from airborne laser scans typically have quite low levels of noise and few outliers. In contrast, point clouds derived from depth maps produced using airborne optical images (occasionally of poor quality) have much higher levels of noise and have large numbers of outliers. Since airborne laser scanning is financially very expensive while airborne optical imaging is many orders of magnitude cheaper, we are motivated to develop analysis and algorithms that can produce accurate segmentation using the information hidden in high noise and masked by many outliers. Our goal in this paper is to draw information for urban reconstruction out of “very bad” data. What we consider here is not just detection of buildings, which can often be performed from a single image but rather extraction—using depth maps with high noise and many outliers derived from several images—of planes (roofs, walls, ground, etc.) for full 3D building reconstruction.

When every 3D point is covered by several images with good radiometry, multi-view dense 3D reconstruction provides building outlines and roof details comparable to those of airborne laser scans. However, computation of depth maps from optical images involves a well-known generic challenge, namely, that it is not always possible to extract reliable depth information in commonly encountered regions of homogeneous color, repetitive texture, low contrast or occlusions. In such situations, there are large numbers of outliers, that is,
points that in ground truth should be on a surface at a given depth but appear in the data set as far away from
the ground truth surface. The contribution of this present paper is robust segmentation methods that can handle
point clouds contaminated not just with a few but with large numbers of outliers.

Progress in segmentation of urban terrain into geometric primitives for analysis of roof detail has been rapid
recently. Local methods, for which only a neighborhood of a point is considered for calculation of a normal vector,
are widely used. Gross et al.\(^5\) compute the gradient of a smoothed elevation map and cluster the values of the
gradient for each building using the \(k\)-means algorithm. One starts with a large number of clusters and reduces
the number of clusters iteratively as long as the angle between the normal vectors corresponding to two clusters
does not exceed a user-defined threshold. Zebedin et al.\(^3\) compute for each 3D point a probability that it lies
in one of several dominant planes and perform nonlocal optimization using a graph cut method\(^6\) or a method
for multi-labeling on Markov random fields.\(^7,8\) However, identification of these dominant planes in point clouds
contaminated by outliers is a non-trivial task. Segmentation of optical images by extending pixel neighborhoods\(^9\)
represents a widely-used alternative that shifts out of the local framework used in the other methods mentioned
above in this paragraph toward a global framework.

Global methods, that is, methods that consider large, non-local portions of the point cloud, have had some
success in reconstruction of surfaces from data with high noise and high number of outliers. RANSAC and the
Hough Transform have successfully been used for roof detail analysis\(^2,10\) by extracting dominant planes from
the whole point cloud. However, when many outliers are present, many points will not be assigned to planes,
which can only partly be compensated by morphological dilatation or the like. Additionally, this process often
produces extraneous planes due to erroneously fitting items that are not local to the phenomenon of interest
or from fitting a local item, such as a V-shaped roof, by a single plane with few inliers that fits neither of the
two roof sections well. The accuracy of global methods discussed above can sometimes be enhanced by making
these methods “less global,” for example, by subdivision of buildings along their natural boundaries.\(^2\) The
enhanced accuracy appears if the subdivisions are “correct.” However, knowledge of the “correct” subdivisions
is tautologically equivalent to knowing a large part of the solution that one is seeking. When the boundaries
of the local regions do not approximate the natural boundaries of the segments well, accuracy is often reduced.
In this paper, we share with many others in the community the desire to build the global result through “less
global” calculations. We seek a way to avoid the tautological problem of effectively having to know in advance
the “correct” boundaries of the local regions, that is, having to know the “correct” segmentation before one
starts to compute the segmentation.

In this paper, we propose a planar segmentation procedure based on \(\ell^1\) regression and compare this procedure
with an analogous planar segmentation procedure based on the widely used RANSAC method.\(^11\) In these
procedures, we first use \(\ell^1\) regression and RANSAC not globally but rather on local neighborhoods to calculate
local normals and then cluster the normals to segment the point cloud into regions that have normals pointing
in the same direction. Then we cluster the physical coordinates of the points in each of these regions into
separate planar segments (roofs, walls, ground, etc.). \(\ell^1\) regression has in the past been recognized as an
alternative to RANSAC. However, direct quantitative comparisons have not previously been carried out and
RANSAC has remained the widespread first choice of the terrain modeling community. Thirty years of experience
with RANSAC in the terrain modeling community have led to excellent practical understanding of RANSAC’s
advantages, including robustness to outliers, and of its disadvantages, including unlimited computing time as well
as occasional ungraceful degradation as the outlier percentage increases. The advantages (robustness to outliers,
graceful degradation and limited computing time) and a disadvantage (need to know the vertical direction) of
\(\ell^1\) regression are also widely understood. This paper complements this understanding with a quantitative
comparison of a RANSAC-based segmentation procedure and an \(\ell^1\)-regression-based alternative.

This paper is structured as follows. After a brief description in Sec. 2 of how depth maps are calculated and of
how the high noise and the outliers in these maps arise, we describe in Sec. 3 the proposed segmentation procedure
based on \(\ell^1\) regression and an analogous procedure based on (local) RANSAC. Computational results for the
\(\ell^1\)-regression-based and RANSAC-based procedures applied to data from Bonnland, Germany and comparative
discussion of these two procedures are presented in Sec. 4. In Sec. 5, we compare \(\ell^1\) regression and RANSAC on
a set of artificial data. Section 6 describes how segmentation results will be integrated into a building modeling
algorithm. Section 7 summarizes the contributions of this paper and outlines future research directions.
2. HIGH-NOISE, OUTLIER-RICH POINT CLOUDS OBTAINED FROM DEPTH MAPS

To see where the high noise and many outliers in depth maps come from, we describe the depth-map process and show data generated from a portion of an aerial video over the village Bonnland in southern Germany.

The first step in generating a point cloud from video is reconstruction of the camera trajectory and of a sparse point cloud by a structure-from-motion algorithm.12 Afterwards, a small subsequence is used to obtain a depth map for a reference image. A detailed description of an algorithm for creating a depth map has been provided by Bulatov et al.13 The algorithm can be summarized as follows. For every pixel $i$ of an image and for discrete values $d$ of depth (the unknown parameter), the value of a cost functional $c(d,i)$ is calculated. The cost functional $c(d,i)$ is aggregated over pairwise cost functionals $c_k(d,i)$, where $c_k$ is, in the simplest case, the absolute difference over gray values of two images of the configuration. This aggregation is robust against occlusion: one can, for example, sum up only low values of $c_k$ instead of summing up all of these values. For each pixel $i$, the “locally calculated” depth is the $d$ that minimizes $c(d,i)$. The cost functional $c(d,i)$ can be used as input for a global smoothing procedure with one of various approaches for multi-labeling on Markov Random fields for 2D graphs6–8 to produce “globally calculated” depth values.

The (global) collection of locally or globally calculated depth values for each pixel is the depth map, from which one calculates 3D points. A reference image for Bonnland, Germany, the result of the structure-from-motion algorithm and two elevation maps, one with 831,020 points calculated by a local method and the other with 846,373 points calculated by a global method, are depicted in Fig. 1. Due to the many homogeneously textured regions in the images, the locally calculated result is very noisy and the global result cannot resolve this problem fully, especially on the streets, as seen in the bottom right of Fig. 1. The depth maps for generating these data were based on five visual images.

Figure 1. Top left: Reference image (one of the video images from the Bonnland sequence). Top right: Result of structure-from-motion algorithm. 3D points are indicated by grey dots. Camera trajectory is indicated by the curve of pyramids. Bottom left: Locally calculated depth map. Bottom right: Globally calculated depth map.

Applying a non-local smoothing algorithm means, essentially, that less reliable depth values (for instance, from pixels in homogeneously textured areas) are overwritten by depth values of their neighbors. Since applying a non-local optimization method requires more memory and computing time, we are interested in algorithms that do not require smoothed, globally calculated data but can accurately segment the high-noise, outlier-rich data generated from locally calculated depth maps.
The procedures that we used to segment the data consist of the following four steps:

Step 1: For each data point $p = (x, y, z)$, calculate the upward-pointing unit normal and a “denoised” data point using planar $\ell^1$ regression or RANSAC on a local neighborhood.

Step 2: Using DBSCAN\textsuperscript{14} or fuzzy $c$-means,\textsuperscript{15} cluster the normals calculated in Step 1 to identify sets of data points that have normals pointing approximately in the same direction. For very large data sets, this clustering may first be carried out individually on separate subdomains and then the clusters on the subdomains combined to create clusters for the whole data set.

Step 3: Use DBSCAN to subcluster the denoised data points in each cluster identified in Step 2. The subclusters correspond to roofs, walls and other connected planar segments of individual buildings.

Step 4: Fit the segments produced in Step 3 by planar $\ell^1$ regression (if $\ell^1$ regression was used on Step 1) or RANSAC (if RANSAC was used on Step 1).

In Step 1, several neighbors of $p$, denoted by $(x_i, y_i, z_i)$ for $i = 1, ..., M$, are selected by the $k$-NN ($k$ nearest neighbor) algorithm\textsuperscript{16} using Euclidean distance in 2D $xy$-space (horizontal distance). In order to fit a plane $z = ax + by + d$ that is robust against outliers, $\ell^1$ regression minimizes the sum of the absolute values of the vertical errors $\sum_{i=1}^{M} |z_i - (ax_i + by_i + d)|$ ($\ell^1$ norm of the error vector), which is a linear program. RANSAC presupposes extraction of several hypotheses of planes through triplets of points and chooses a plane with the largest set of inliers. Calculation of the upward-pointing unit normal $(-a, -b, +1)/\sqrt{a^2 + b^2 + 1}$ of this plane and of the “denoised data point” $(x, y, ax + by + d)$ concludes this step.

DBSCAN and fuzzy $c$-means were chosen as the clustering algorithms for Steps 2 and 3 for the following reasons. DBSCAN clustering, which is used on Step 2 (optionally) and Step 3 (always), forms clusters by searching for subsets that are locally dense. DBSCAN requires the user to choose two parameters, namely, $\varepsilon$, the radius of a local neighborhood of each point, and the minimum number of points that is needed to form a cluster. For a given positive integer $N$, fuzzy $c$-means (used optionally on Step 2) finds $N$ clusters by assigning a normal to the cluster in which the normal’s membership function is largest. Like the more commonly used $k$-means clustering,\textsuperscript{5} fuzzy $c$-means clustering is a distance-based clustering method. However, fuzzy $c$-means is more flexible than $k$-means and was therefore chosen instead of $k$-means as one of the clustering algorithms on Step 2. Fuzzy $c$-means generates clusters that cover all points of the data set and leaves no points in an “unknown” category outside of the clusters that are identified. DBSCAN produces clusters that typically have smaller spreads than the clusters produced by fuzzy $c$-means. Also, DBSCAN allows a category “unknown” to arise, which is meaningful in general as well as necessary for nonplanar regions such as vegetation. DBSCAN and fuzzy $c$-means have different and largely complementary advantages and disadvantages, which was the reason for including both of them as options in Step 2.

In Step 3, we use DBSCAN only and do not use fuzzy $c$-means because it is essential in this case to have a category “unknown”. The regions produced in Step 2 include many small, scattered subregions that result from the high noise level in the data and do not correspond to physical roofs or other segments of interest. These subregions need to be excluded from the subclustering process in Step 3. DBSCAN accomplishes this. Fuzzy $c$-means would require all points to be part of a roof or other segment and would not accomplish the objective of this step, namely, to identify connected planar segments of significant size.

On Step 4, we use the same method ($\ell^1$ regression or RANSAC) that was used on Step 1. It is possible that a combination of $\ell^1$ regression on one step and RANSAC on the other would produce better output than either alone. Research on that is under consideration for the future. At present, using the same method on Step 4 that was used on Step 1 provides a comparison in which all of the differences can be directly associated with $\ell^1$ regression and RANSAC.
4. IMPLEMENTATION DETAILS AND COMPUTATIONAL RESULTS
FOR BONNLAND DATA

The high-noise, outlier-rich data set for Bonnland, Germany described in Sec. 2 is an excellent data set for
testing the $\ell_1$-regression-based and RANSAC-based segmentation procedures described in Sec. 3. Other urban
terrain data sets, for example, those produced using LIDAR, have low noise and few outliers—but at much higher
financial cost. Segmentation of low-noise, few-outlier data sets is of interest in its own right. However, our goal
here is to provide procedures for cheap, “bad” data rather than for expensive, good data.

Before presenting the computational results, we describe here some implementation details. The experiments
were run on a MATLAB 2012a platform. On Step 1, for each point $p = (x, y, z)$, we identified a neighborhood
of 400 points using the $k$-NN algorithm implemented in the Matlab module knnsearch. On Steps 1 and 4,
$\ell_1$ regression was implemented using the MATLAB package CVX with default parameters. RANSAC was
implemented using the RANSAC Toolbox for MATLAB with a chi squared probability threshold for inliers,
a false alarm rate (probability of never picking a good minimal sample set) and the standard deviation of the
RANSAC-internal noise set to 0.99, $10^6$ and 0.01, respectively. To carry out Step 2, we divided the convex hull
of the horizontal $xy$ coordinates of the data points into a tensor product of $3 \times 2$ rectangular subdomains. For
DBSCAN on these subdomains, we chose, based on computational parameter studies, different $\epsilon$ and minimum
numbers of points in the subdomains so that the sizes of the clusters were physically relevant. For fuzzy $c$-means,
we used $2, 10^{-5}$ and 1000 as the exponent for the matrix $U$, the minimum amount of improvement and
the maximum number of iterations, respectively. The number of fuzzy $c$-means clusters $N$ was 4, 5 or 6. The
software for DBSCAN was the Java module ELKI and, for fuzzy $c$-means, it was the MATLAB module fcm. To
create clusters for the whole domain from the clusters for the $3 \times 2$ subdomains, we combined subdomain clusters
if over 50% of two clusters overlapped. There is no quantitative guidance for this portion of the procedure and it
was accomplished by human judgement. The parameters for DBSCAN in Step 3 were chosen by trial and error
such that each subclustering (wall, roof or ground) was as close to the true underlying structure as possible.

The plots for dominant planar segments generated using these parameters are shown in Fig. 2. To generate
these results, the $\ell_1$-regression-based procedure required 3737 sec and 3642 sec to generate the information on
the top left (fuzzy $c$-means on Step 2) and the bottom left (DBSCAN on Step 2), respectively, of Fig. 2 and the
RANSAC-based procedure required 5073 sec and 4976 sec to generate the information on the top right (fuzzy
$c$-means on Step 2) and the bottom right (DBSCAN on Step 2), respectively, of Fig. 2. These computing times
consist mainly of the time required by $\ell_1$ regression and RANSAC on Step 1, which allows one to estimate that
$\ell_1$ regression is, in this context, over 25% faster than RANSAC.

On Step 2, most of the clusters of normals produced by DBSCAN have spans of less than $60^\circ$. In contrast,
fuzzy $c$-means produces clusters that are quite broad, including clusters, the normals of which differ in direction
by $90^\circ$ or more, which is not physically meaningful for normals nominally representing the direction of a planar
segment. Nevertheless, for the Bonnland data, fuzzy $c$-means produces segmentations that are more accurate
than those produced by DBSCAN. This may be due to some specific properties of the high-noise data. This
issue is not yet understood and will be the object of future investigation.

In order to evaluate the computational results in the context in which they will be used in the future, we go
one step further into the process of the building roof reconstruction. We consider only those segments that lie
within buildings; the building maps are obtained by height thresholding of the elevation map and by neglecting
segments that are too small or too narrow. We process these segments by morphological operations (closing and
opening) and polygonize them. The polygons are filtered by their magnitude and eccentricity in order to eliminate
segments that are too small or too narrow (the latter are often roof ridges that degrade the reconstruction), see
Fig. 3, top. To each polygon is attributed the plane equation from the original segment with which it has the
largest intersection set. Finally, pairs of neighboring segments are determined and the cutlines, that is, the
intersections of planes, are calculated. The projection of the cutline of two planes $\pi_1$ and $\pi_2$ into an image has a
representation

\[ l = (Pv_1) \times (Pv_2) \]

in homogeneous coordinates, where $P$ is a $3 \times 4$ projection matrix that links the world coordinate system with the
image coordinate system and $v_i$ denotes the $i$-th column of the orthogonal complement to the vector subspace.
spanned by \( \pi_1 \) and \( \pi_2 \). If such a cutline lies close to both polygons (within a corridor of 15 pixels), we set up its endpoints to match the extent of the polygons and plot it in Fig. 3, bottom.

By analyzing cutlines, we can draw conclusions about the segmentation results. Missing cutlines are an indication of under-segmentation while extraneous cutlines are typical for over-segmentation. Cutlines that do not coincide with characteristic edges in the foreground of the image indicate that the plane parameters were estimated with insufficient accuracy (typical for under-segmentation). We see in Fig. 3 (first and second rows) that, for example, in the L-shaped building, all five important cutlines were determined after performing the procedure based on \( \ell_1 \) regression and clustering with either DBSCAN or fuzzy \( c \)-means. Four of the cutlines were reconstructed correctly while one cutline does not coincide completely with a characteristic edge in the building outline. This means that a nearly correct reconstruction of the roof structure of this building is possible. The cutline that does not coincide completely is the line between the segments labeled 1 and 2 in Fig. 3, top left. The reason for the incorrect cutline is under-segmentation. In reality, Segment 2 consists of two segments with nearly the same normal vectors but different plane equations. Due to this under-segmentation, a slightly wrong set of plane parameters was assigned to the smaller segment resulting in deviation in the corresponding cutline. Nearly parallel planes with only slight offset are a situation that makes it clear that clustering by normal vectors only is insufficient and subclustering of the data points as in Fig. 2 is needed.

No over-segmentation has occurred in Fig. 3, top left, in spite of the fact that several cutlines in other buildings of the figure have not been captured. The \( \ell^1 \)-regression-based procedure combined with clustering by fuzzy \( c \)-means yields more correctly reconstructed cutlines (cf. the building in the bottom right corner of Fig. 3, second row on the right) and, at the same time, several over-segmentations. The RANSAC based-procedure produces much worse results when points are clustered with the DBSCAN method; not a single meaningful cutline could be reconstructed (third row). The results of the RANSAC-based procedure combined with clustering by fuzzy \( c \)-means are comparable with those produced by \( \ell^1 \) regression. The plane equations produced by \( \ell^1 \) regression seem to be slightly more accurate since the corresponding cutlines coincide better with the characteristic edges in the images; however, the segmentation results are less accurate. For example, Segment 1 is too large at cost

Figure 2. Top left: Dominant segments generated by \( \ell_1 \) regression on Steps 1 and 4 with DBSCAN clustering of normals on Step 2 and DBSCAN clustering of denoised data points on Step 3. Top right: Dominant segments generated by RANSAC on Steps 1 and 4 with DBSCAN clustering of normals on Step 2 and DBSCAN clustering of denoised data points on Step 3. Bottom, left and right: same as top, but with fuzzy \( c \)-means clustering of normals on Step 2.
Figure 3. Top left: Roof polygons for segmentation data generated by $\ell_1$ regression on Steps 1 and 4 with DBSCAN clustering of normals on Step 2 and DBSCAN clustering of denoised data points on Step 3. Second row, left: Roof polygons for segmentation data generated by $\ell_1$ regression on Steps 1 and 4 with fuzzy $c$-means clustering of normals on Step 2 and DBSCAN clustering of denoised data points on Step 3. Third and forth row, left: The same as in the first two rows, but with RANSAC used on Steps 1 and 4. On the right, top to bottom: cutlines obtained from the segmentation result on the left projected onto the original image.
of Segment 2 in the bottom left image of Fig. 3, which results in a cutline that is too short. Summarizing, the results produced by \( \ell^1 \)-regression are superior to those produced by RANSAC and the former are less sensitive to the clustering method.

The two buildings with characteristic segments labeled 3 and 4 in Fig. 3, second row, right, are worth a short note: The cutlines of these buildings are slightly inside the edges of the roofs. This is correct because the facades in this village are not at the edges of the roofs but slightly behind those edges. This indicates that the default procedure for modeling facades, namely, inserting vertical trapezoids with arms that connect two neighboring vertices of the roof polygon, can be improved and, in the future, better localization of facades will be possible even in noisy point clouds.

5. RESULTS FOR ARTIFICIAL DATA

In addition to carrying out the computational experiments with real data described above in Section 4, we also carried out computational experiments with artificial data representing a roof with two planes at inclinations of 45° to the horizontal. The 101 \( \times \) 51 points of the basic data set with no noise (the “no-noise roof”) are points \((x,y,z)\) where \(x = i - 50\) and \(y = j - 25\) for \(i = 0, 1, \ldots, 100\) and \(j = 0, 1, \ldots, 50\) and where \(z = 50 + i\) for \(i = 0, 1, \ldots, 50\) and \(j = 0, 1, \ldots, 50\) (“Segment 1”) and \(z = 150 - i\) for \(i = 51, 52, \ldots, 100\) and \(j = 0, 1, \ldots, 50\) (“Segment 2”). Let \( \nu \) be a \( N(0, 1) \) Gaussian random variable or a Student \( t \) random variable with 2 degrees of freedom (df). The Student \( t \) distribution with 2 df is very heavy-tailed, that is, it has large spread and many outliers, and thus provides noise that forms a significant challenge for segmentation procedures. The data sets used in the computational experiments described below consisted of 101 \( \times \) 51 points \((x,y,z+c\nu)\) where \(x, y\) and \(z\) are the coordinates of points on the no-noise roof described above, \(\nu\) is Gaussian or Student \( t \) noise and \(c = 0, 0.1, 0.3, 0.6, 1, 2, 3, 10\). Because of the high-noise, outlier-rich point clouds produced from depth maps described above in this paper, we have particular interest in large \(c\) and in \(\nu\) coming from a Student \( t \) distribution.

The normals were calculated as on Step 1 of the procedure stated at the beginning of Section 3. The local neighborhood used in Step 1 consisted of the \(M = 317\) points (fewer when the point \(p\) is near a boundary) for which the horizontal coordinates \((x,y)\) are at a radius of 10 or less from the \((x,y)\) coordinates of the reference point \(p\). In the data sets under consideration, the horizontal distance between adjacent data points is 1. For RANSAC, the chi squared probability threshold for inliers, the false alarm rate (probability of never picking a good minimal sample set) and the standard deviation of the RANSAC-internal noise were set to 0.99, \(10^{-6}\) and \(0.01\), respectively. Steps 2–4 of the procedure of Section 3 were not carried out.

We determined whether the normals computed by \( \ell^1 \)-regression and by RANSAC correctly identified the segment to which the points belong by computing whether the directions of the computed normals were sufficiently close to the directions of the ground truth normals. If the cosine of the angle between the computed normal of a point and the ground truth normal for that point was \(\geq 0.99\) (angle \(\leq 8.1096^\circ\)), the point was classified as being correctly segmented. Otherwise, the point was classified as incorrectly segmented.

Tables 1 and 2 provide the percents of correctly classified points and the computing times for the \( \ell^1 \)-regression-based and RANSAC-based procedures. As indicated in these tables, RANSAC produces better normals than \( \ell^1 \) regression for small noise (\(c = 0.1\) and 0.3). However, for all other cases—the cases with higher noise and more outliers that are of particular interest in this paper—\( \ell^1 \)-regression produces much better normals than RANSAC. For huge noise (\(c \geq 2\)), RANSAC fails completely while \( \ell^1 \)-regression still produces excellent results that degrade gracefully as \(c\) increases. The computing time of the \( \ell^1 \)-regression-based procedure is roughly constant for all noise levels. As the noise increases from low (\(c = 0.1\)) Gaussian noise to huge Student \( t \) noise, the computing time of the RANSAC-based procedure skyrockets from 1.6 times to 110 times the computing time of the \( \ell^1 \)-regression-based procedure. These results suggest, as do the results for the Bonland data in Section 4, that urban-terrain research based on RANSAC should be complemented by research based on other robust computational methods such as \( \ell^1 \)-regression.

6. INTEGRATION INTO A FULL BUILDING RECONSTRUCTION PROCEDURE

In urban terrain, buildings are the most common items of interest and how one can integrate the \( \ell^1 \)-regression-based and RANSAC-based planar segmentation procedures described above into a full building reconstruction
Table 1. Percent of correctly classified points ("%") and CPU times (τ) in seconds for calculation of normals for data with Gaussian noise

<table>
<thead>
<tr>
<th>Method</th>
<th>c = 0.1</th>
<th>c = 0.3</th>
<th>c = 0.6</th>
<th>c = 1</th>
<th>c = 2</th>
<th>c = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ℓ¹ regression</td>
<td>93.7</td>
<td>837</td>
<td>93.1</td>
<td>864</td>
<td>93.0</td>
<td>832</td>
</tr>
<tr>
<td>RANSAC</td>
<td>100.0</td>
<td>1364</td>
<td>99.5</td>
<td>9809</td>
<td>82.3</td>
<td>27165</td>
</tr>
</tbody>
</table>

Table 2. Percent of correctly classified points ("%") and CPU times (τ) in seconds for calculation of normals for data with Student t noise

<table>
<thead>
<tr>
<th>Method</th>
<th>c = 0.1</th>
<th>c = 0.3</th>
<th>c = 0.6</th>
<th>c = 1</th>
<th>c = 2</th>
<th>c = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ℓ¹ regression</td>
<td>93.1</td>
<td>912</td>
<td>92.9</td>
<td>932</td>
<td>92.9</td>
<td>922</td>
</tr>
<tr>
<td>RANSAC</td>
<td>100.0</td>
<td>2378</td>
<td>92.9</td>
<td>11163</td>
<td>41.6</td>
<td>51404</td>
</tr>
</tbody>
</table>

procedure is of importance. It is beyond the scope of this paper to actually carry out this integration but the path to such integration is known and we summarize it here. The interested reader is encouraged to consult the article by Gross et al., where a detailed description of a procedure for airborne-laser-generated point clouds is presented. Laser-generated point clouds typically consist of high-quality points while the point clouds of interest in this paper consist of low-quality points obtained from (local) depth maps. Fortunately, versions of the procedure that are suitable for points obtained by photogrammetric methods have recently been developed for filtering out vegetation, for computation of Digital Terrain Models (DTMs) and for roof detail analysis.

The full procedure consists of two main parts, namely, building detection and building reconstruction. For building detection, the difference of the elevation map and the (not necessarily geo-referenced) DTM is computed. In the majority of cases, regions of large difference between the elevation map and the DTM correspond to either buildings or trees. The next step involves separation of vegetation. The last step of the building detection part discards components that are too small or too narrow, after which the remaining components (building hypotheses) are labeled and the building reconstruction part of the procedure is applied to each of them. To reconstruct a building starting from the height-thresholded elevation map, its 2D outline must first be modeled in form of a polygon (optionally rectangular). Then roof detail analysis is performed. The results of the ℓ¹-regression-based or RANSAC-based planar segmentation procedure identify dominant roof segments within 2D building outlines. These segments are polygonized and the intersections of corresponding planes with each other and with the building outlines are calculated. Building walls are in general not seen from above in good resolution, so, at present, they are modeled as indicated at the end of the previous section. The results presented in Sec. 4 and Sec. 5 suggest that the ℓ¹-regression-based and RANSAC-based planar segmentation procedures are good candidates for sources of information not merely about roofs and other areas seen at non-glancing angles but also about areas with higher levels of noise such as walls. After roofs and walls have been reconstructed, texturing of the building polygons with the content of the input images (after occlusion analysis) concludes the building reconstruction procedure.

7. CONCLUSIONS AND OUTLOOK

A common strategy nowadays is to use RANSAC-based procedures in high-noise, outlier-rich cases. The evidence presented in this paper suggests that there is much to be gained by further investigation not only of RANSAC-based procedures but also of ℓ¹-regression-based procedures. For the Bonnland data, the ℓ¹-regression-based procedure produces results that are somewhat better in accuracy and in computing time than those produced by the RANSAC-based procedure. For the Bonnland data, the ℓ¹-regression-based procedure is also less sensitive to the clustering method. For the artificial data, ℓ¹ regression performs much better than RANSAC in both accuracy and computing time. These results suggest significant further potential for ℓ¹ regression.

The computational results presented in Sec. 4 are of lower absolute accuracy than many other urban-terrain-reconstruction results in the literature because the other results are typically based on “good data” (low noise,
few outliers) and ours are based on “bad data” (high noise, many outliers). Both “good data” and “bad data” will continue to be available. However, it is likely that the balance between the good data and the bad data will shift in favor of the bad data because of cost. The analysis and algorithms presented here show that one can produce accurate segmentation using the information hidden in high noise and masked by many outliers.

The reader will have noticed that we do not do any pre-processing of the data by smoothing. Standard smoothing procedures, typically based on identification and rejection of outliers and on averaging the remaining data points, perform well for low-noise data that represent smoothly varying objects but do not perform well for data with many outliers or for data that represent objects with discontinuities (such as urban terrain). Step 1 of the \( \ell^1 \)-regression-based and RANSAC-based procedures includes denoising of the data, which can be considered to be a type of smoothing. However, in contrast to standard smoothing, which smears discontinuities, this denoising preserves discontinuities.

\( \ell^1 \) regression and RANSAC are not limited to reconstructing planar segments but are suitable for reconstructing segments with more complicated, nonplanar structure. It is therefore expected that extensions of the \( \ell^1 \)-regression-based and RANSAC-based procedures will be capable not merely of identifying roofs, walls and ground but also of identifying vegetation and other nonplanar features in natural and man-made terrain.

In their present versions, the \( \ell^1 \)-regression-based and RANSAC-based procedures do not use information about the quality of points. However, confidence values that measure the reliability of the depth values can be calculated. The procedures can be adjusted to make use of these confidence values. The characteristics (for example, discretization of depth levels, bias and potential asymmetric noise distributions) can also be modeled and these models used to improve the accuracy of the procedures. Future comparisons that include this additional information may include not merely \( \ell^1 \) regression and (local) RANSAC but also other methods such as global RANSAC or the J-linkage method of Toldo and Fusiello.

ACKNOWLEDGMENTS

The work of Jian Luo and Zhibin Deng was generously supported by US Army Research Office (ARO) Grant #W911NF-04-D-0003. It is ARO policy that research personnel do not need to do joint work with ARO personnel in order to receive grants from the Army Research Office.

REFERENCES


[10] Vosselman, G. and Dijkman, S., “3D Building model reconstruction from point clouds and ground plans,” 


Intelligence 28(8), 1–13 (2006).


in Learning and Control], 95–100 (2008).


terrain reconstruction from m(uav)-videos for geo-information applications,” Unmanned Aerial Vehicle in

from aerial images,” ICPRAM 2, 275–280 (2012).

Kim, S.-J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R.,