PERCEPTUAL ORGANIZATION OF 3D SURFACE POINTS

Impyeong Lee, Toni Schenk

Dept. of Civil and Environmental Engineering and Geodetic Science
The Ohio State University, Columbus, OH 43017, USA
lee.1517@osu.edu, schenk.2@osu.edu

Commission III, WG III/3

KEY WORDS: Photogrammetry, Vision Sciences, Organization, Segmentation, Surface, LIDAR, Grouping, Points

ABSTRACT:

Perceptual organization is proposed as a promising intermediate process toward object recognition and reconstruction from 3D surface points, which can be derived from aerial stereo-images, LIDAR data or InSAR data. Here, perceptual organization is to group sensory primitives originating from the same object and has been emphasized as a robust intermediate-level grouping process toward object recognition in human and computer vision. Despite intensive research on 2D data, perceptual organization of 3D entities is still in its infancy, however. Therefore, the purpose of this research is to develop a robust approach for constructing perceptual organization particularly with irregularly distributed 3D surface points. The scope of perceptual organization presented in this paper is limited to signal, primitive and structural levels. At the signal level, we organize raw 3D points into spatially coherent patches. Then, at the primitive level, we merge the patches into co-parametric surfaces. Finally, at the structural level, we group the surfaces into perceptually meaningful surface clusters. We establish a novel approach and implement the approach as an autonomous system. The system is evaluated with real LIDAR data by inspecting the quality of organized output. The evaluation substantiates a promising performance of the system. The organized output serves as a valuable input to higher order perceptual processes, including the generation and validation of hypotheses in object recognition tasks.

1. INTRODUCTION

A far-reaching goal of digital photogrammetry is to reconstruct the world with an abstract description generated from various sensory inputs as autonomously as possible. This inversion problem is ill-posed and the solutions are usually based on introducing assumptions about the 3D object space and by applying suitable constraints. It has long been recognized that surfaces play an important role in the quest of reconstructing scenes from sensory data such as images. Surfaces are predominantly represented (measured) by irregularly distributed 3D points. Such a point cloud can be derived from aerial imagery (stereopsis), LIDAR data or InSAR data.

Perceptual organization deals with grouping sensory inputs that originate from the same object by finding structural regularity from or imposing structural organization on the inputs. It has been recognized as a crucial component that makes human perception powerful and versatile. In computer vision, perceptual organization is often used as a robust intermediate-vision process toward object recognition.

Sarkar and Boyer (1993) propose a classificatory structure of perceptual organization based on the dimension over which an organization is sought and the abstraction level of features to be grouped. The structure has two axes: one axis denotes 2D, 3D, 2D plus time and 3D plus time; and the other axis represents signal, primitive, structural and assembly levels. For example, segmentation of surfaces from a 3D point cloud is classified into 3D signal level perceptual organization. In addition, further grouping of segmented surfaces is categorized into 3D primitive or structural level perceptual organization.

Previous work in perceptual organization concentrated on 2D organization, dealing with all the abstraction levels and emphasizing the structural level. In 3D organization, most previous studies address only the signal level, particularly focusing on range image segmentation (Koster and Spann, 2000; Jiang et al., 2000; Liu and Wang, 1999; Hoover et al., 1996). The need of perceptual organization in various levels of 3D will increase because 3D sensors have become cheaper and more readily available. Boyer and Sarkar (1999) conclude that perceptual organization in 3D is one of the most important research directions in this area.

Therefore, the purpose of the research reported in this paper is to develop a robust approach for constructing perceptual organization, particularly with irregularly distributed 3D surface points. In the interest of brevity we focus on the most salient features of the proposed approach. The interested reader may find missing details in Lee (2002).

2. PROPOSED APPROACH

A conceptual framework for computing perceptual organization from 3D surface points is described by Lee and Schenk (2001). Under this framework, we have developed a novel approach that is based on a bottom-up (or data driven strategy) and thus less application-dependent although it slightly favors urban applications.

We now summarize the overall process. From a given set of irregularly distributed 3D surface points we compute perceptual organization at the signal, primitive and structural levels. At the signal level, we organize the raw points into spatially coherent surface patches. Then, at the primitive level, we merge the patches into co-parametric surfaces and identify surface boundaries. Finally, at the structural level, we group the surfaces into perceptually meaningful surface clusters and derive intersections, corners, and the ground surface from them.

2.1 Signal Level Process

At the signal level, we organize raw 3D points into spatially coherent patches. The signal level process involves establishing and refining adjacency among points, selecting seed patches, and growing patches from the seeds. Each patch is identified with the interior points and the parameters of a plane approximated to the points. The input and output of the process are summarized in Table 1.

Input	• A set of points.
Output	• A set of patches. • A point adjacency graph.

Table 1: Input and output of the signal level process.

2.1.1 Point Adjacency Adjacency is based on the distance between points. Five types of adjacency are introduced. First, '2D' adjacency is assigned to any pair of points if the horizontal distance between the points is less than a given threshold. For a 2D adjacent pair, '3D' adjacency is assigned if the 3D distance is also less than the threshold and '2D only' adjacency is assigned otherwise. 3D adjacent pairs are further classified into '3D refined' if they pass a refining process and '3D unnecessary' otherwise.

The refining process is to identify some unnecessary ones from 3D adjacent pairs. For example, a point on the top of a car is undesirably identified to be adjacent to other point on the ground once the distance between them is not so large. Such unnecessary adjacent pairs can be identified if other pairs within the same area are statistically consistent except them. The refining process is summarized as:

1. Select a point as a center and collect all 3D adjacent points.
2. Fit a plane to the collected points using a robust estimator such as Least Median Squares Estimator (Koster and Spann, 2000).
3. Classify the points into inliers and outliers with respect to the robustly fitted plane.
4. If the center is an outlier, eliminate the adjacency between the center and every other point. Otherwise, eliminate the adjacency between the center and every outlier.
5. Repeat step 1 to 4 for every point.

Based on the adjacency assignment, we establish a point adjacency graph, where a node incorporates a point; an arc links a pair of adjacent points, retaining the type of the adjacency.

2.1.2 Seed Patches Seed patches serve as initial patches from which planes start to grow. Hence, seed patches should be as *complete* and *homogeneous* as possible. Here, completeness implies that all the actual planes should grow from seed patches; homogeneity indicates that seed patches should satisfy planar surface models. To accomplish completeness and homogeneity, we propose the following selection procedure:

1. For every point in a data set, generate a small patch by clustering a fixed number of neighboring points.
2. Fit a plane to each patch and compute the fitting error.
3. Establish an ordered-list (heap) of the seed patches so that the patch with the minimum fitting error will be fetched first.

2.1.3 Patch Growing Patches are growing from seed patches, followed by verification. The growing process comprises four main tasks:

1. Fetch a non-corrupted seed patch.
2. Find the nearest point to the current patch.
3. Perform a hypothesis test to examine whether the point is statistically consistent with the current patch.
4. After passing the acceptance test, the current patch is updated accordingly.

The first task keeps fetching a seed patch from the seed heap until a non-corrupted one is found, where a corrupted seed is one that contains one or more points that have already been assigned to another patch during the previous growing processes. Also, an arc heap is initially created with the arcs of 3D refined types in the point adjacency graph, which link at least a point of a seed patch. This heap is constructed so that the arc of the minimum length will be fetched first, where the length is defined as the distance between two points linked by the arc. The second task finds the nearest point to the current patch among the external points adjacent to at least an internal points based on the 3D refined adjacency. An arc in the arc heap connects two internal points, or one internal point and one external point. Hence, one should repeatedly pop the shortest arc from the heap until an arc connected to an external point is found. Then, the external point is the nearest point to the patch. If the heap becomes empty, or the length of the shortest arc exceeds a threshold, the growing process terminates. The third task statistically determines whether the nearest point is consistent with the current patch based on a hypothesis test. The null hypothesis is "the point is on the surface" while the alternative hypothesis is "the point is not on the surface". The last task updates the current estimates of the plane parameters by incorporating the new point into the current model using a sequential least squares estimator.

After growing each patch, we verify it in terms of its size, roughness, and geometry. If one of these criteria is violated, the patch is discarded and its points are reset to *unsegmented* status so that they can be segmented to other patches.

2.2 Primitive Level Process

At the primitive level, we generate surfaces by merging some segmented patches. The primitive level process involves constructing patch adjacency graphs, computing merging confidence of adjacent patches, merging adjacent patches of high merging confidence into surfaces, and deriving surface boundaries. Each surface is identified with the interior points, the surface parameters, and the surface boundaries. The input and output of the process are summarized in Table 2.

Input	• A set of patches.
Output	• A set of surfaces. • A set of surface boundaries. • A patch/surface adjacency graph.

Table 2: Input and output of the primitive level process.

2.2.1 Patch Adjacency Graph A patch adjacency graph includes segmented patches as nodes and adjacency between the patches as arcs. Adjacency between two patches are assigned if at least a point of a patch is adjacent to a point of the other patch. Based on the types of the point adjacency, the patch adjacency retains three different types such as '2D', '3D', and '2D only'. The 2D only adjacency is assigned to two patches that are adjacent to each other only in 2D. For example, the 2D only adjacency can be established between a roof patch of a building and its neighboring ground patch. The 2D only adjacency is particularly important since it often suggests the existence of a missing vertical patch between 2D only adjacent patches.

2.2.2 Merging Confidence Adjacent patches are to be merged if there exists a significant evidence that they share the same surface *parameters* and *roughness*. The merging confidence represents the degree of confidence in merging two patches. The merging confidence between two patches $\mathcal{P}_1$ and $\mathcal{P}_2$ is defined as:

$$\theta_{merge}(\mathcal{P}_1, \mathcal{P}_2) = \theta_{merge}(\mathcal{P}_1 \leftarrow \mathcal{P}_2) \cdot \theta_{merge}(\mathcal{P}_2 \leftarrow \mathcal{P}_1)$$

where $\theta_{merge}(\mathcal{P}_1, \mathcal{P}_2)$ is the confidence in merging $\mathcal{P}_1$ and $\mathcal{P}_2$; $\theta_{merge}(\mathcal{P}_1 \leftarrow \mathcal{P}_2)$ is the confidence in merging $\mathcal{P}_2$ into $\mathcal{P}_1$; $\theta_{merge}(\mathcal{P}_2 \leftarrow \mathcal{P}_1)$ is the confidence in merging $\mathcal{P}_1$ into $\mathcal{P}_2$. $\theta_{merge}(\mathcal{P}_1 \leftarrow \mathcal{P}_2)$ is defined as the p-value determined by a test statistic of a statistical test. This test examines whether the points of $\mathcal{P}_2$ are consistent with those of $\mathcal{P}_1$ in terms of the parameters of the surface fitted to the points and the associated fitting errors. The surface model can be primitively a plane model but easily extended to more complex surfaces such as quadratic surfaces.

2.2.3 Merging Patches Two adjacent patches with high merging confidence are merged into one surface. The merging process is summarized as follows:

1. Construct or update the patch adjacency graph.
2. Compute the merging confidence between every pair of adjacent patches.
3. Select the pair of the highest merging confidence.
4. If the confidence of the selected pair is greater than a threshold, merge the patches and go to step 1.
5. Otherwise, quit the merging process.

The patch adjacency graph finally updated after the merging process is considered as the surface adjacency graph.

2.2.4 Surface Boundaries Surface boundaries represents a general shape for a cluster of points included by each merged surface, being computed from interior points based on the α-shapes (Edelsbrunner et al., 1983) algorithm. Here, the α-shapes are generalizations of the convex hull of a point set. They are constructed as subgraphs of the Delaunay triangulation. Parameter α controls the level of the shape details.

2.3 Structural Level Process

At the structural level, we organize the merged surfaces into perceptually meaningful surface clusters. The process involves identifying surface adjacent boundary edges, computing various perceptual cues such as *connectedness*, *continuity*, *parallelism* and *elevatedness*, constructing the graph associated with each cue, hypothesizing intersections and corners, grouping surfaces into meaningful surface clusters, and identifying ground surface clusters and above-ground polyhedral structures. Table 3 summarizes the input and output of the process.

Input	• A set of surfaces. • A set of surface boundaries. • A surface adjacency graph.
Output	• A set of surface clusters. • A set of intersections. • A set of corners. • A surface adjacent boundary edge set. • A surface connectedness graph. • A surface continuity graph. • A surface parallelism graph. • A set of ground surface clusters. • A set of above-ground structures.

Table 3: Input and output of the structural level process.

2.3.1 Adjacent boundary edges If boundary edges are adjacent to boundaries of other surfaces they become *adjacent boundary edges*. Their identification is very useful at the structural level because some important attributes such as connectedness and elevatedness are based on them.

2.3.2 Surface Connectedness

Surface connectedness is defined as a random variable indicating the confidence in the hypothesis that two surfaces are connected. This retains *continuous* and *non-continuous* types, according to the changes of the surface normals near their adjacent boundaries. The non-continuous types are further classified into *convex* and *concave* types. Connectedness with a type is computed from every pair of the 3D adjacent surfaces identified from the surface adjacency graph.

2.3.3 Surface Continuity

Continuity indicates how smoothly two surfaces can be connected. Although two surfaces are not connected, they can be continuously connected through an imaginary intermediate surface between the surfaces. Then, the continuity between the surfaces is high while the connectedness is very low. This is the main difference from connectedness. In addition, continuously connected surfaces show high continuity but non-continuously connected surfaces show low continuity.

2.3.4 Surface Parallelism

Parallelism is defined as a random variable indicating the confidence in the hypothesis that two surfaces are parallel. It is associated with the similarity of the surface normals, the closeness of surfaces in the normal direction, and the area of the overlap of the surfaces.

2.3.5 Surface Elevatedness

Elevatedness is an attribute of a surface that indicates how much elevated a surface is than adjacent surfaces along the surface boundaries. Elevatedness is a good indicator to identify ground surfaces from a set of surfaces since a ground surface is usually less elevated comparing to its adjacent surfaces. For example, the average elevation of a ground surface can be higher than a roof of building. However, if we compare elevations only along adjacent boundaries, the ground surface is much lower. Hence, elevatedness is a promising attribute for the identification of ground surfaces.

2.3.6 Perceptual Cue Graphs

Based on connectedness, continuity, and parallelism computed from a set of surfaces, we establish surface connectedness, continuity, and parallelism graphs.

2.3.7 Intersections

An intersection can be hypothesized by every pair of two surfaces connected non-continuously. This hypothesis is then confirmed if the straight line intersected by two surfaces is close to the adjacent boundaries of the surfaces. The ending points of the confirmed intersection are deliberately determined along the straight line.

2.3.8 Corners

A corner can be derived by every set of three surfaces connected non-continuously. To identify such sets, we derive so-called *tri-arcs* from the connectedness graph, where a tri-arc is assigned to three nodes connected to each other. A tri-arc invokes a corner hypothesis. This hypothesis is confirmed if the corner is close to the adjacent boundary edges of the three surfaces.

2.3.9 Grouping Surfaces

Surfaces are grouped into surface clusters so that all the surfaces originating from the same object (ex. at least the ground) is organized into the same cluster. The grouping criteria is designed to cluster two surfaces that retain high connectedness and low elevatedness between them. This design is based on two observations, that is, 1) highly connected surfaces must belong to the same object; 2) vertical discontinuities hardly exist between the surfaces originating from the ground. Every arc from the connectedness graph provides a surface pair to be examined for grouping with the grouping criteria. The grouping algorithm is summarized as follows:

1. Establish a *union-find* structure where every surface is assigned to a separate cluster. This structure supports efficient operations of *union* of two clusters and *find* the cluster of a surface.
2. Push all the arcs of the connectedness graph into a *heap*, where the heap stores the arc of the highest connectedness at the head.
3. Pop the arc of the highest connectedness from the heap and identify the two surfaces linked by the arc.
4. If the type of the connectedness between the surfaces is concave, go to step 9.
5. If the connectedness is less than a threshold, go to step 10.
6. Find two surface clusters including each surface. If they are the same, go to step 9.
7. Compute elevatedness between the clusters. If its absolute value is is greater than a threshold, go to step 9.
8. Union of the two clusters.
9. Repeat steps 3 to 8 until no more arcs remain at the heap.
10. Identify the cluster assigned to every surface using the union-find structure.

2.3.10 Ground Surface Clusters

We compute the elevatedness and the area for every surface cluster and identify the cluster of the lowest elevatedness and the largest area as ground surface clusters.

2.3.11 Above-ground polyhedral structures

Above-ground polyhedral structures such as buildings may retain concave connectedness and significant elevatedness in a structure. Hence, after excluding the clusters identified as the ground, we further group the surface clusters with relaxed criteria comparing to those in section 2.3.9. Then, each cluster is identified as an above-ground polyhedral structure.

3. IMPLEMENTATION AND EXPERIMENTS

3.1 Implementation

We implemented the proposed approach as an autonomous system that generates a three-level perceptual organization from 3D surface points. The system consists of three cascaded subsystems, where the subsystems produce signal, primitive and structural level organizations, respectively.

We attempted to fully incorporate the strategies of Object Oriented Programming (OOP) for developing the system software. The entire software was thus programmed with ANSI C++ with Standard Template Library (STL). The software involves many newly defined classes corresponding to main objects and algorithms of the system, such as points, edges, patches, surfaces, surface clusters, graphs, LMS estimation, LMeDS estimation and others. The software has been compiled and tested mainly under Microsoft Windows Operating Systems (OS), but it is platform independent.

The system is then theoretically analyzed in terms of parameter selection and computation complexity. The analysis indicates that the system is robust to parameter selection and maintains moderate computational complexity. Lee (2002) provides further details.

3.2 Experiments

The main test areas are the sub-sites of the Ocean City test site. A more detailed description of this test site is presented by Csatho et al. (1998). The data sets, acquired by a LIDAR system, cover many urban areas in Ocean City. We tested the proposed system with many sets and presented the results of two sets among them. The main properties of these sets are summarized in Table 4. Set A is selected to illustrate the full system processes; and set B is used to demonstrate the overall quality of the output over a large area. Figure 1 and 2 show the perceptual organization generated from set A and B, respectively.

Set	No. points	Area $[m^2]$	Density $[points/m^2]$
A	4633	5564	0.83
B	32493	32926	0.99

Table 4: Properties of the test data.

4. CONCLUSIONS

We constructed a novel approach that computes perceptual organization at three levels from 3D surface points and implemented this approach as an autonomous system. This system was tested with real LIDAR data sets of various characteristics. The system performance was evaluated by inspecting visually the quality of the organized output. This evaluation strongly demonstrates the usefulness of the proposed approach.

The proposed approach produces autonomously with moderate computation loads, robust, explicit, complete, computationally efficient and hierarchical descriptions from raw surface points. The organized output thus serves as a valuable input to higher order perceptual processes, including the generation and validation of hypotheses in object recognition tasks.

Future research will concentrate on the following topics:

- To evaluate rigorously the system performance through quantitative analysis as well as qualitative inspections, with the input data of various ranges and characteristics.
- To develop a mechanism that adjusts the system based on domain knowledge specific to a given input and a pursuing application.
- To apply the output to higher level processing such as DEM generation, building reconstruction, change detection, urban modelling and other object recognition and reconstruction tasks.

References

Boyer, K. L. and Sarkar, S., 1999. Perceptual organization in computer vision: status, challenges, and potential. *Computer Vision and Image Understanding,* 76(1), pp. 1–5.

Csatho, B., Krabill, W., Lucas, J. and Schenk, T., 1998. A multisensor data set of an urban and coastal scene. In: *International Archives of Photogrammetry and Remote Sensing, Object Recognition Scene Classification From Multispectral and Multisensor Pixels*, Columbus, OH, USA, Vol. 32, Part 3/2, pp. 26–31.

Edelsbrunner, H., Kirkpatrick, D. G. and Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory,* 29(4), pp. 551–559.

Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A. and Fisher, R. B., 1996. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 18(7), pp. 673–689.

Jiang, X., Bunke, H. and Meier, U., 2000. High-level feature based range image segmentation. *Image and Vision Computing,* 18(10), pp. 817–822.

Koster, K. and Spann, M., 2000. MIR: an approach to robust clustering-application to range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22(5), pp. 430–444.

Lee, I., 2002. Perceptual organization of surfaces. Ph.D. dissertation, The Ohio State University, Columbus, OH, USA.

Lee, I. and Schenk, T., 2001. 3D perceptual organization of laser altimetry data. In: *International Archives of Photogrammetry and Remote Sensing, Land Surface Mapping and Characterization Using Laser Altimetry*, Annapolis, MD, USA, Vol. 34, Part 3/W4, pp. 57–65.

Liu, X. and Wang, D. L., 1999. Range image segmentation using a relaxation oscillator network. *IEEE Transactions on Neural Networks,* 10(3), pp. 564–573.

Sarkar, S. and Boyer, K. L., 1993. Perceptual organization in computer vision: a review and a proposal for a classificatory structure. IEEE Transactions on Systems, Man and Cybernetics 23(2), pp. 382–399.

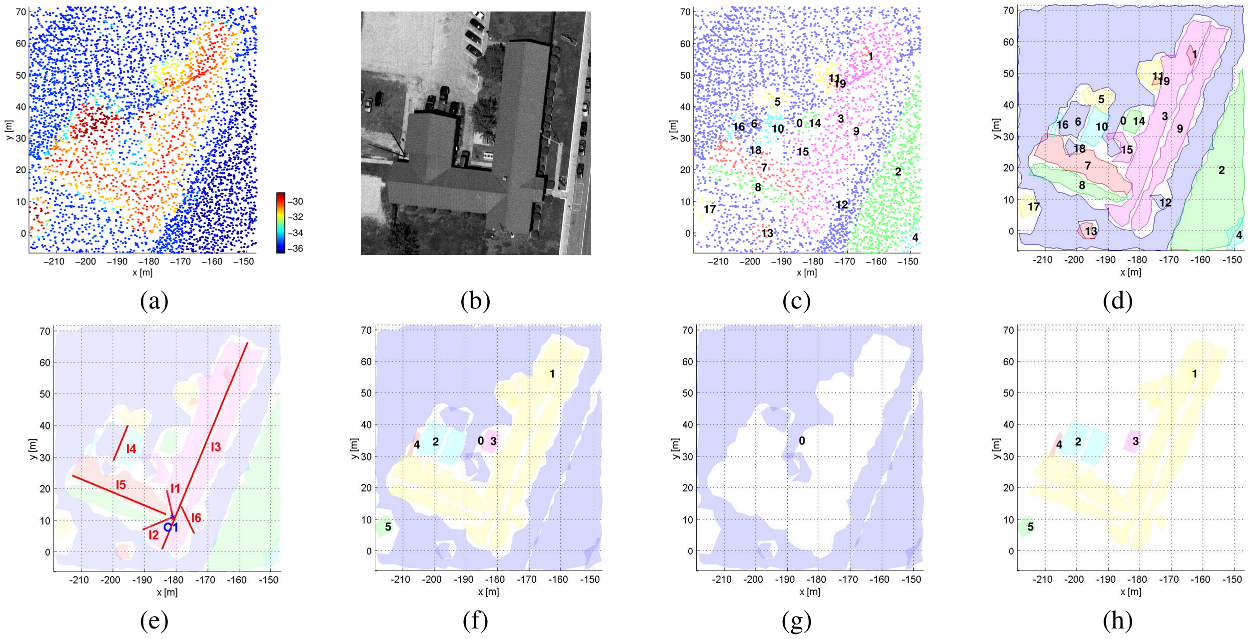

Figure 1: Perceptual Organization of Set A: (a) 3D surface points with the elevations encoded by the colors. (b) Aerial photo of the same area. A building, a parking place and a road are identified. (c) Segmented patches visualized with six different colors. (d) Merged surfaces with boundaries. (e) Derived intersections and corners. (f) Grouped surface clusters. (g) Identified ground surface cluster. (h) Extracted Above-ground surface clusters.

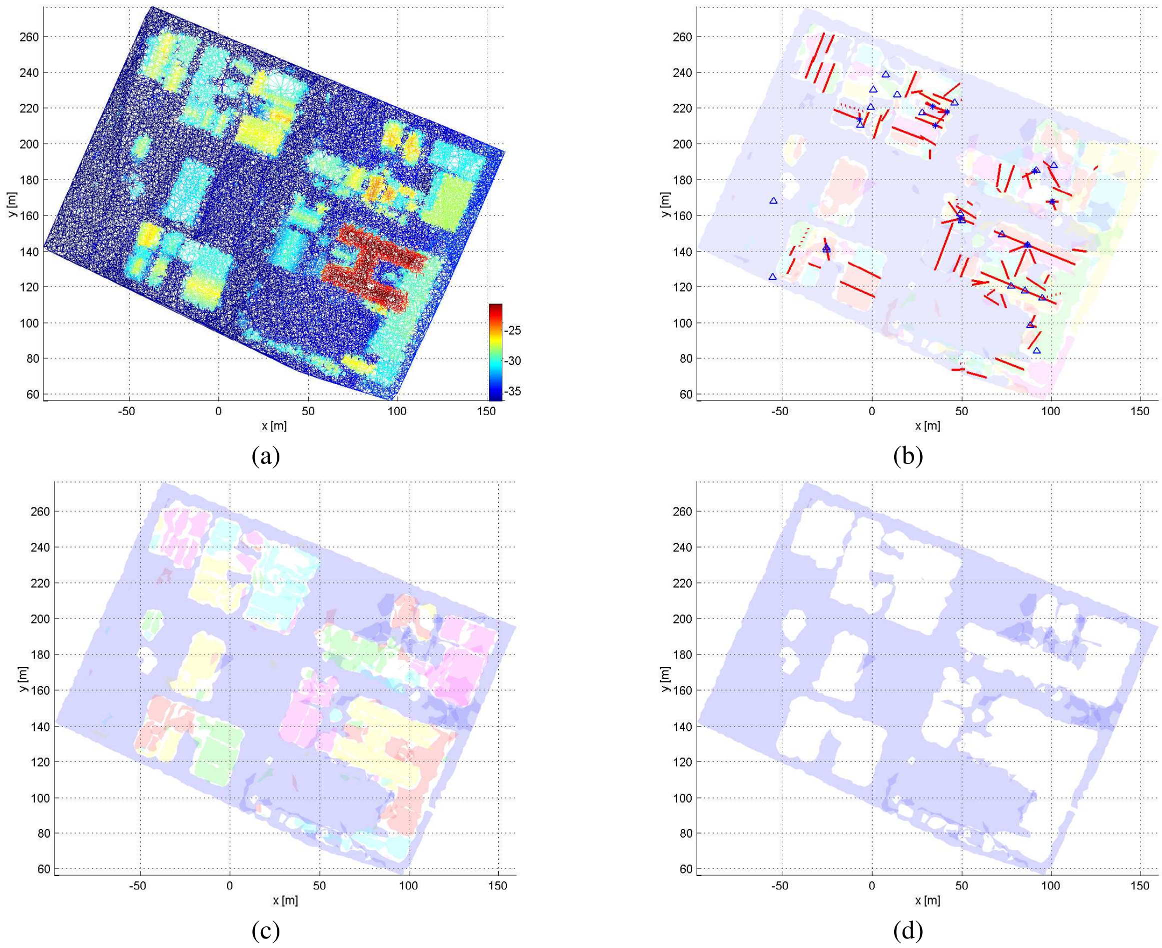

Figure 2: Perceptual Organization of Set B: (a) 3D surface points. (b) Merged surface with boundaries, and derived intersections and corners. (c) Grouped surface clusters. (d) Identified ground surface cluster.