Chapter 4 – FROM POINT CLOUDS TO 3D MODELS

Effective modelling of 3D urban terrains by assimilation of sensed point cloud data is challenging and an active area of research. Typically the sensed objects and environment require models of complex geometry and topology which cannot be formulated in terms of common functional surfaces. Urban features such as tree canopies, overpasses, utility lines, elevated walkways, tunnels, and building interiors (Figure 4-1) require more general 3D representations that allow multi-valent height fields and higher genus topologies.

a)

b)

c)

d)

e)

f)

Figure 4-1: Examples of Urban Terrain Including: a) Building passages; b) Overpasses; c) Buildings and archways with complicated topology; d) Industrial complex of machinery, piping and utilities; and e)-f) Man-made structures with passageways, vehicles, and vegetation in Bonnland (left) and Norrköping (right) where data sets were contributed by SET-118 members.

Military applications require timely processing of large data sets. The processing framework should be able to:

i) Efficiently update the fitted geometry as new point cloud data is received;

ii) Naturally incorporate multi-scale versions of the fitted surfaces;

iii) Locally encode meta information; and

iv) Identify geometric features in a form suitable for database organization to enable fast queries for feature classification and matching such as those described in Section 4.3.

Additional applications may require real-time line of sight calculations of viewsheds for mission planning and autonomous navigation. In this chapter, requirements for representation are detailed and a brief review of the approach of both explicit and implicit methods is presented, along with their respective advantages and disadvantages. In Section 4.2, the pros and cons of cube and tetrahedral-based hierarchical representations are described in terms of the requirements expected of the representations. Section 4.3 provides a detailed description of classification of urban geometric objects. Section 4.4 illustrates a range of sample point cloud data sets which were contributed by NATO SET-118 member organizations and subsequently processed

using various assimilation algorithms to reconstruct approximations to the sensed surfaces. This is followed in Section 4.5 by three examples of model instantiations by processing 3D sensed data collections generated by other electro-optical sensors: 3D road extraction using stereo image pairs (Section 4.5.1), building reconstruction from multi-aspect high-resolution InSAR data (Section 4.5.2), and the use of photo interpretation for height estimation in terrain maps (Section 4.5.3).

4.1 CONCEPTS AND REQUIREMENTS

Representations for 3D urban terrains which hold the most useful potential for supporting military operations such as those outlined in previous sections, should have a framework which incorporates the following features:

a) Representations of the geometry and the topology of the terrain should be accurately assimilated directly from highly non-uniformly distributed point-cloud data, preferably without human intervention. Issues such as missing data should be addressed and reflected in the representations.

b) Representations used in fitting the point cloud should provide compression ratios of up to 100:1 using natural geometric primitives. Encoding/decoding of terrain should be efficient, scale linearly with the data, and use representations for efficient bitstream transmission in a client-server framework.

c) Computational effort of the processing algorithms and their derivative products, such as classification and line of sight, should scale linearly with the data. Moreover, algorithms should be amenable to parallelization.

d) Metrics appropriate for geometric modelling and applications should be naturally incorporated into the surface fitting algorithms. Rate-distortion estimates, i.e., trade-offs between compression and metric-accuracy should be provided for common classes of urban terrain. Local estimates of quality of fit should be available as local metadata.

e) Multi-resolution versions of the terrain representations should provide models with increasing geometric and topological detail as a user drills down locally.

f) Representations should incorporate dynamic learning or re-learning of 3D urban terrain to efficiently update models as supplementary or replacement data becomes available.

g) Representations should provide fast line-of-sight and viewsheds for complex 3D terrains.

h) Data structures used in processing should be hierarchical and naturally formulated to allow efficient look-up and search algorithms, both locally and through multiple resolutions.

We provide a several examples to illustrate the significance of each of these requirements. The first example (see Figure 4-2) is a 2.5D LIDAR data collection provided by US Army Corps of Engineers Topographical Engineering Center derived from a flyover of Baltimore, MD. Since the data has been processed to lie on a uniformly spaced grid with pixel values determining a height map, the surface may be considered as a univalent functional surface with values at a pixel position determined by the intensity.

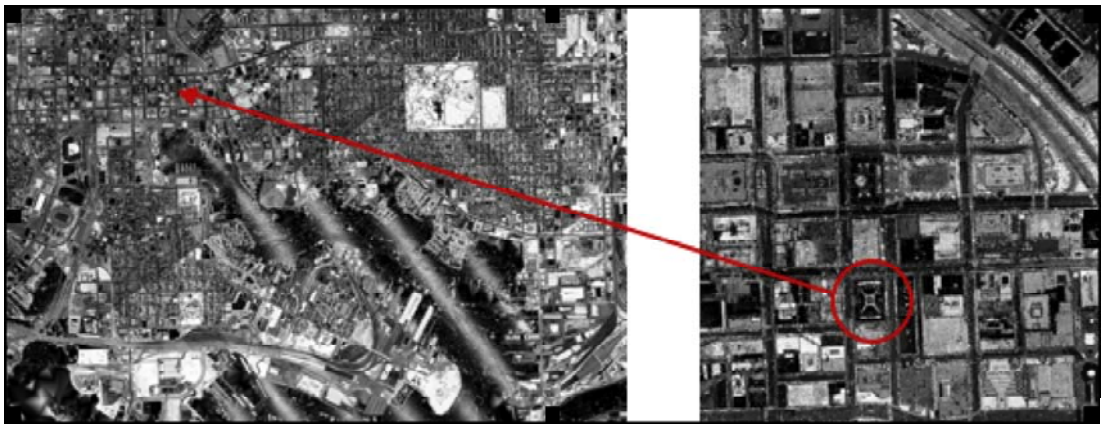


Figure 4-2: Requirement of Multi-Scale Localization. A rendering of a portion of a Baltimore LIDAR data set as an image (left) with a zoomed view in the vicinity of the domed building (right).

Although this point cloud does not have the full character of topology and geometry as might be collected by other more exhaustive LIDAR imaging, it still exhibits properties of large scale data collections over large coverages that require hierarchical processing methods in order to quickly provide views with varying resolutions at user-defined locations. One representative local drill-down around a domed building (circled in red in Figure 4-2) is provided in Figure 4-3a) by zooming in to a high resolution of the point cloud. In this image the colouring of points is according to height for assist in providing perspective for visualization purposes only. Due to the pose of the sensing platform the sides of the buildings are typically obscured and do not provide LIDAR returns, which results in missing data from the vertical walls in a full 3D model. If care is not taken in the surface reconstruction to respect data sparsity and identify regions with missing data, then over fitting of the data is likely to result, as illustrated in Figure 4-3e).

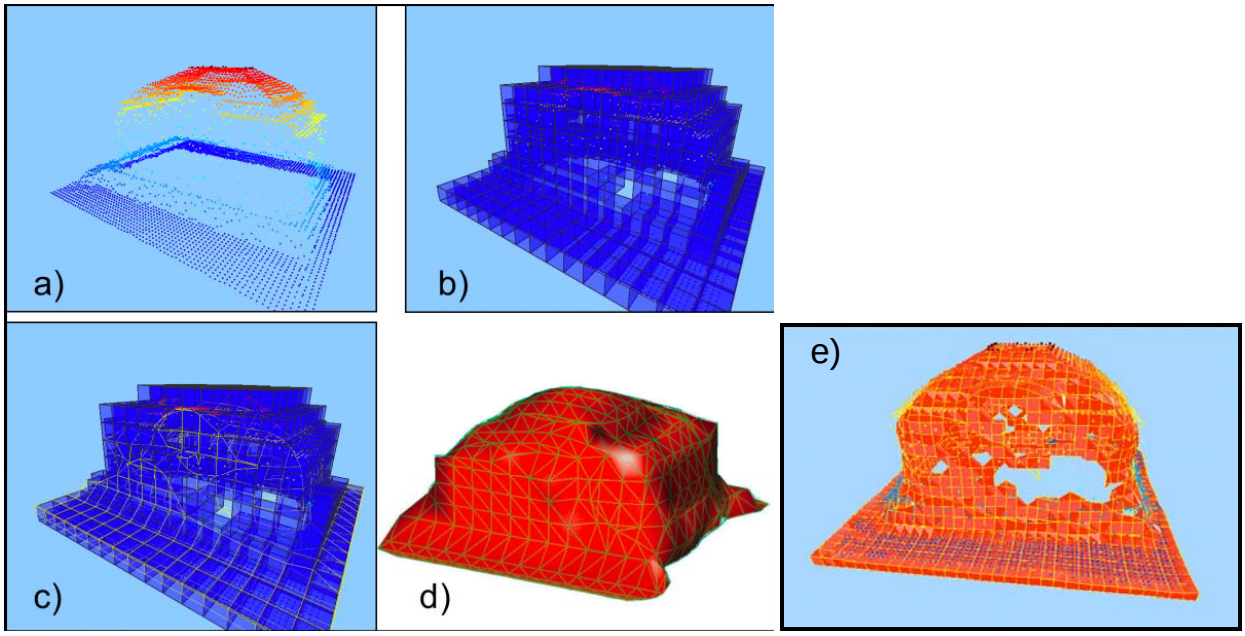


Figure 4-3: Domed Building Circled in Figure 4-2: a) Zoomed view of the point cloud in the vicinity of a domed building, indicating missing data obscured by airborne collection; b) Local oct-cubes; c) Curves indicating implicit distance field to point cloud interpolated from values on vertices of local oct-cubes; d) Fitted surface provided by implicit level surface for distance field; and e) Effect of over-resolving the oct-cubes and selecting a level set parameter too small for the data resolution which results in overfitting the data.

As we have remarked, the raw Baltimore LIDAR data collection was processed in order to provide an estimated height value for each pixel coordinate. For many applications the data may be required to be assimilated with updates of redundantly sampled regions with the hope of providing better estimation (see Figure 4-4).

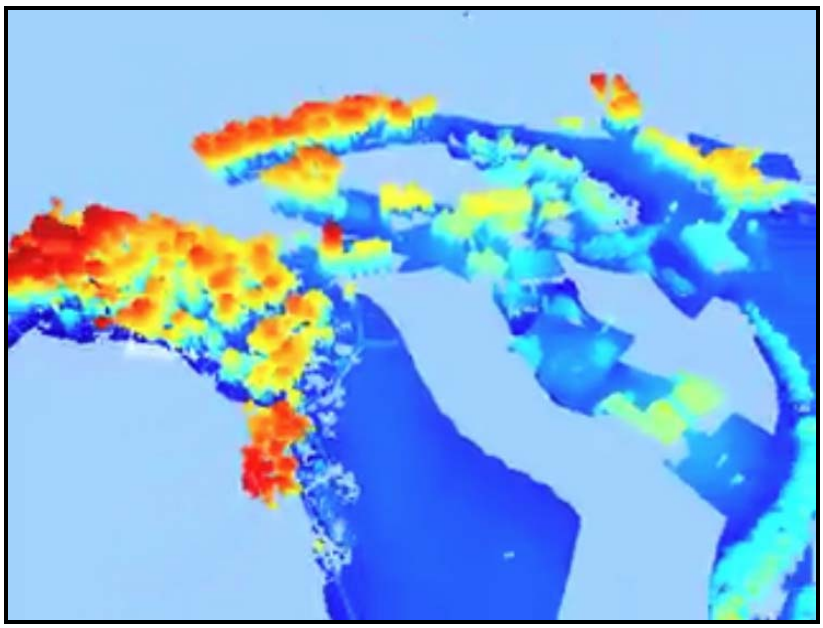


Figure 4-4: Eglin[1] LIDAR Flyover with Overlapping Swaths.

Other applications, such as autonomous navigation of unknown or modified terrain, require on-line adaptive assimilation of the terrain. Figure 4-5 illustrates on-line learning of a 2.5D functional surface in mid-acquisition of the overflight LIDAR data pictured in Figure 4-4. Statistical estimation of surface heights should be updated as the overlapping regions are acquired.

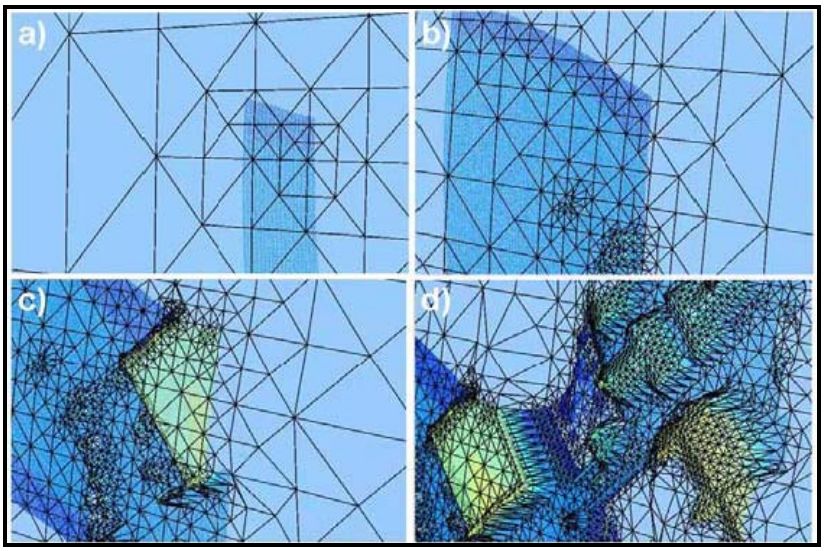
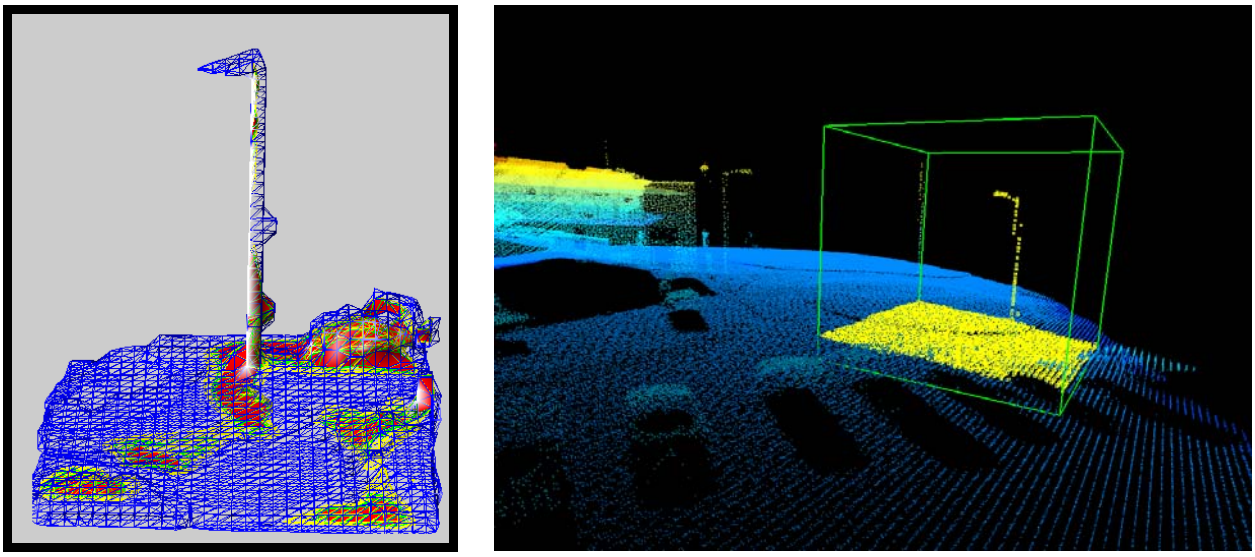


Figure 4-5: Image Sequence of Adaptive Sub-Division Builds a) – d) for On-Line Surface Reconstruction from Eglin[1] LIDAR Overflight Data.

[1] AFRL/MNG VEAA Data Set #1 of Eglin AFB, with permission.

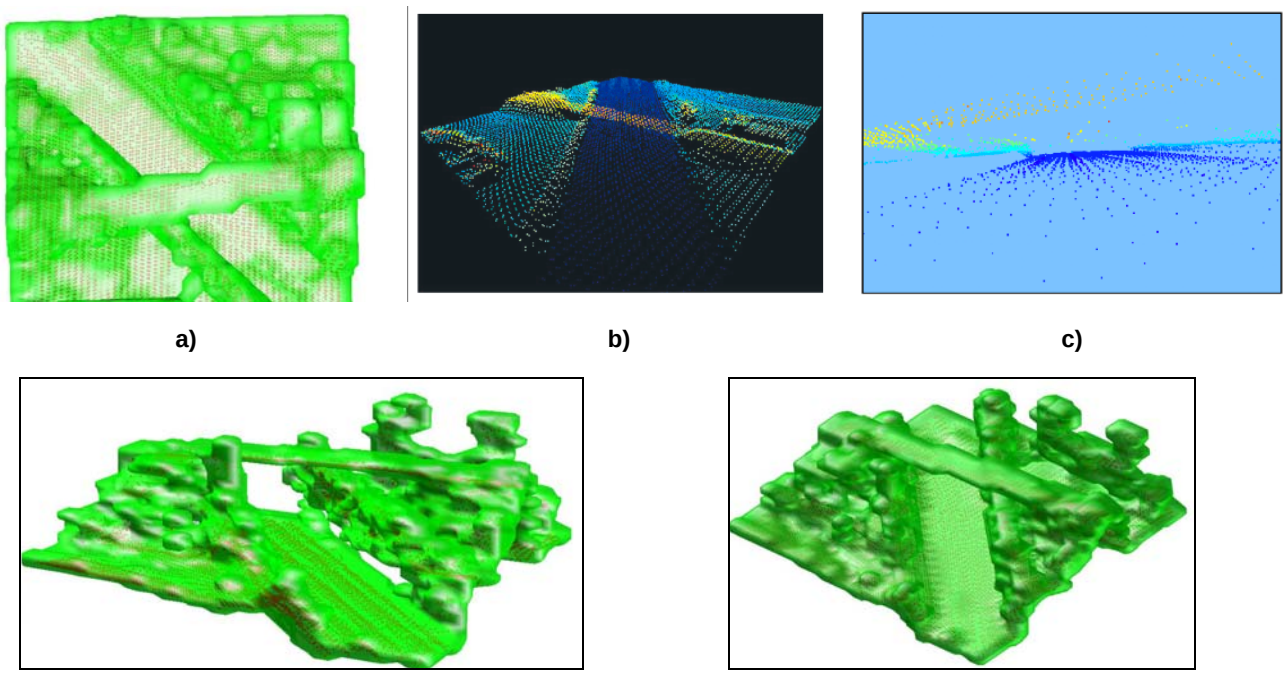
Although the reconstructed geometry in Figure 4-5 provides a coarse approximation of the urban terrain and is satisfactory for some applications, finer resolution is available and may be finely modelled as shown in Figure 4-6. This is an example of simple topology, but non-functional 3D univalent surface.



a) b)

Figure 4-6: a) Close Up of Parking Lot Light Pole from Eglin LIDAR Overflight Data. Missing data can also be observed; and b) Fine Scale Surface Reconstruction Using Level Sets is Used to Fit the Point Cloud Data and is Able to Resolve Additional Detail.

To provide an example involving somewhat more complicated topology, we use publically available overflight LIDAR data prepared by the International Society for Photogrammetry and Remote Sensing Commission III – Photogrammetric Computer Vision Working Group [100]. The data shown in Figure 4-7a) was obtained by modifying these data so as to extend the roadway underneath the overpass to fill in the occluded portion. The resulting data has similar sensor characteristics, but is a genus 1 topological object with more complicated geometry due to the vegetation near the intersection.

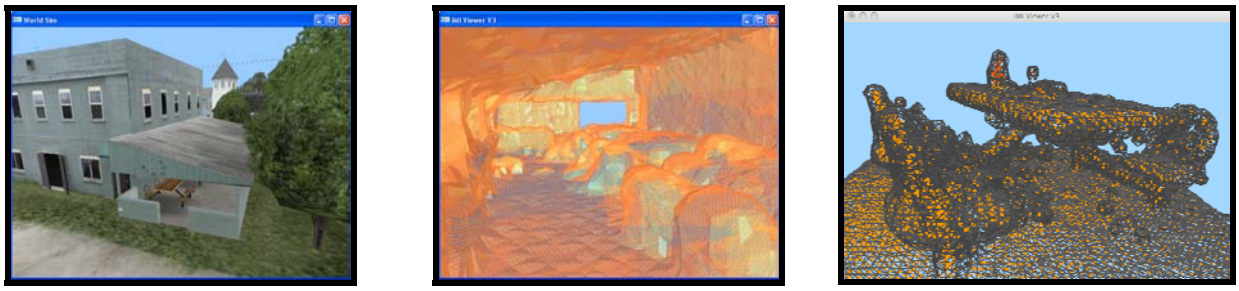


a) b) c)

d) e)

Figure 4-7: a) Original LIDAR Data for Underpass; b) Close Up View from a Different Perspective than the Imaging Sensor, in particular, Looking Under the Underpass; c) Top View of Intersection; d) Reconstructed Surface of Underpass and Neighbouring Vegetation and Trees; and e) Alternate View Showing Overfitting of the Sparse Data Along the Roadway Borders.

A more complicated topology and geometry is shown in Figure 4-8 in which the reconstructed surfaces of synthetic flash LIDAR are rendered. This data was generated by the South Carolina LIDAR simulator using a sequence of ten flash images.

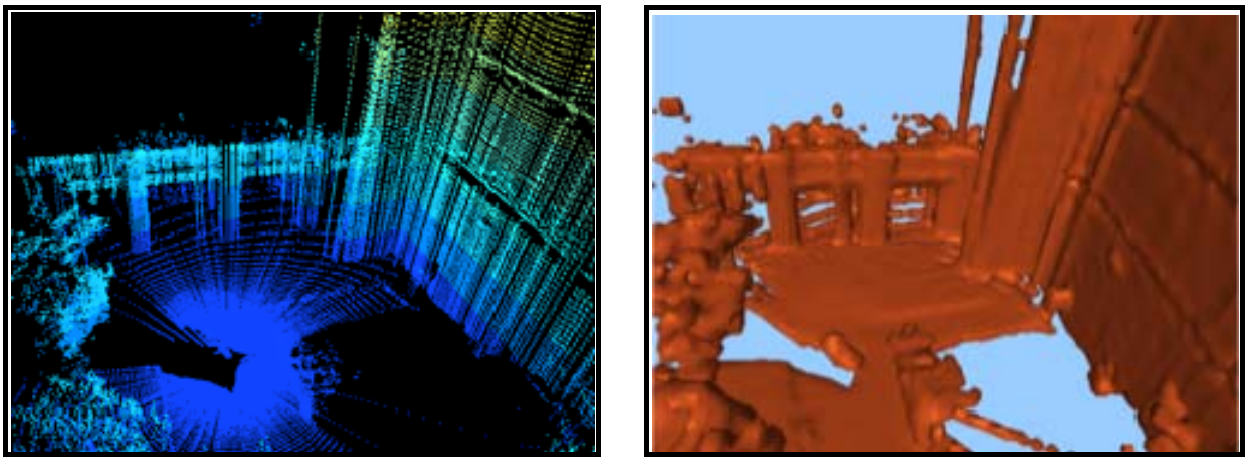


a) b) c)

Figure 4-8: a) Simulated Flash LIDAR Data of Patio Area – Ft. Benning; b) Coarse Learned Surface from Point Clouds Resulting from 10 Registered 128 x 128 Flash LIDAR Images; and c) Finer Resolution of Isosurface of Point Cloud of Table Constructed Using Unsigned Distance Field from Point Cloud in b).

The final example of this section illustrates LIDAR data generated in more typical situations by ground-based platforms. In this case a line scanning Hokuyo LIDAR is mounted on a vertically oriented bar which is rotated about the axis of the bar to produce a relatively dense point cloud of a court yard near Hancock Hall on the campus of Virginia Polytechnic Institute and State University in Blacksburg, Virginia, USA. This data exhibits many of the problems described earlier in this section. In particular, the topology and

geometries are complex and difficult to assimilate into a high quality, reliable surface fit (Figure 4-9). Difficulties arise in large part due to ambiguities of missing or occluded data since information of the acquisition process, such as time-stamped geolocation of the sensor, is not utilized.



a) b)

Figure 4-9: a) Raw LIDAR Data Generated by Hokuyo Line Scanning Sensors Which Were Rotated at a Fixed Position about a Vertically Aligned Axis; and b) Surface Reconstruction Using a Level Set of an Unsigned Distance Captures Some Important Topology, Such as Building Faces and Columns, and Demonstrates Complexities Arising Both from the Sensed Surfaces as well as Processing Artefacts.

4.2 MODEL REPRESENTATIONS

Surface modelling of scattered, highly heterogeneously distributed point clouds presents many challenges. Previously the approaches to fitting geometric structure to this type of data can be classified as either *explicit* or *implicit* methods. The selection of which method is used typically depends upon the end applications and the properties of the point cloud. In this section a brief description of each methodology is presented, followed by examples and discussion of the pros and cons. Ideally, models for terrain surfaces should seamlessly incorporate both explicit and implicit representations into hybrid models, exploiting the advantages of both of these viewpoints while minimizing their drawbacks.

4.2.1 Explicit Surface Representations

An explicit representation usually takes the form of a parametric representation of the surface. For 2.5D surfaces the parameterization is simply of the form $z = f(x,y)$ where z is height at position (x,y). In terrain applications, however, more intricate parameterizations are required in order to model urban terrains with overhanging structures, open building entrances and interiors, vegetation, and similar features. These more general parameterizations take the form $(x(u,v),y(u,v),z(u,v))$ where (u,v) ranges over the parameter space. A simple example of a non-functional parameterized surface would be that of a sphere of radius ρ:

$$\left(\rho\, \boldsymbol{sin}(u)\boldsymbol{cos}(v), \rho\, \boldsymbol{sin}(u)\boldsymbol{sin}(v), \rho\, \boldsymbol{cos}(u)\right), \quad \text{where } 0 \le u \le \pi, 0 \le v \le 2\pi .$$

The departure here from usual terrain models (DTED, DEM) is that explicit representations are taken to be locally the graph of a function in *some* coordinate system which is not assumed to be the standard (x,y,z) Euclidean system. This draws on methods and concepts from computer aided geometric design (CAGD) which involves invaluable, but computationally expensive, processing tools in industrial design [47].

Various research journals such as Elsevier's *Computer Aided Geometric Design* are dedicated particularly to the mathematical and algorithmic issues of this subject. These local coordinate systems are typically estimated by the local structure of the point cloud by determining the local density and variation of the point cloud. The most natural and efficient approach to determine local structure is to *divide and conquer* using hierarchical coarse-to-fine partitioning of the data into occupancy cells. This also has the advantage of representing the data at different scales and allows for local statistics to be easily computed and accessed. Two methods of partitioning into occupancy cells are described later in this sub-section: octrees of cubes and bin-trees of tetrahedra. Various criteria can be used to determine when there has been sufficient partitioning to estimate a local orientation (or projection) for which the surface can be represented locally as a functional surface. A common popular approach to determine the orientation is to use the covariance matrix and corresponding principal component analysis associated with that segment of the point cloud. Once an orientation is estimated for a local segment, a similarity transformation is applied to rotate back to Euclidean coordinates (x,y) so that standard surface fitting methods, using splines for example, can be used. The resulting surface fit is then rotated back to the original local coordinate system as a local parameterized surface fit of the point cloud. These local versions are all stitched together into a global parameterized representation of the desired form $(x(u,v),y(u,v),z(u,v))$.

Next we describe two methods to quickly analyze and cluster a 3D point cloud into occupancy cells.

4.2.1.1 Octree Representation

An octree is a hierarchical tree data structure. It consists of a cubic root node that is recursively sub-divided into 8 child nodes of increasing resolution at each level of the tree. It is a structure similar to a 3D grid but with the added advantage of being multi-resolution. Figure 4-10 shows an example of an octree with corresponding tree representation, where levels are represented with different shades of blue.

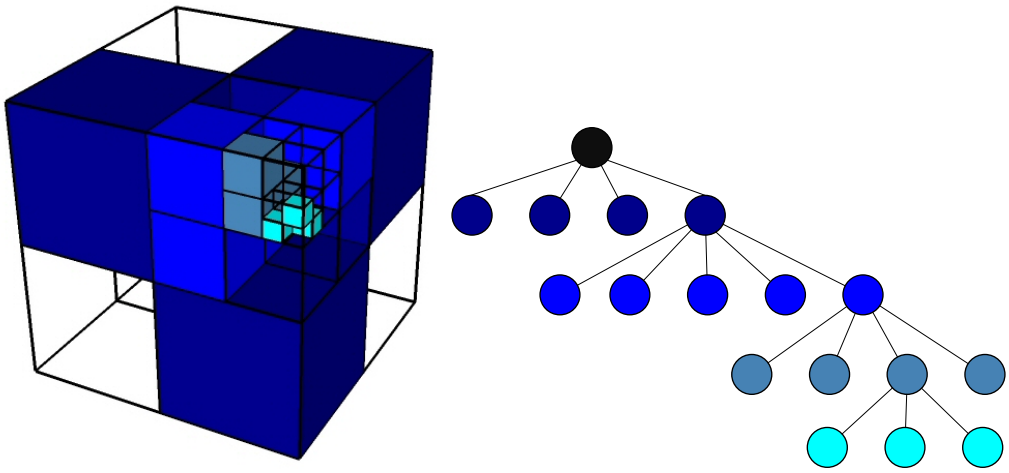


Figure 4-10: Octree with the Corresponding Tree Representation.

Octrees are often used to optimize and accelerate the display of 3D point clouds. In this case, a root node whose size is sufficient to contain the point cloud is recursively sub-divided into child nodes until the number of points contained by each leaf node is below a threshold. This type of representation eases operations such as ray tracing.

Octrees may also be used as dynamic structures to store the occupancy state of the environment as it is measured [48]. For this application, each node is assigned the occupancy probability of the volume it encompasses. As new data is acquired, new nodes are created on demand and existing nodes see their

occupancy probability updated. The maximal resolution of the octree (size of terminal nodes) is selected in accordance to the required model precision.

One advantage of octrees is that nodes are only created for parts of the environment where information is available, incidentally lowering memory requirements. For comparison, when using a 3D grid, one has to generate all the cells even if some of them are unused. In addition, octrees may be compressed easily to lower their resolution and memory footprint. On the other hand, octrees have a more complex structure and navigation between nodes might prove challenging for some applications.

4.2.1.2 Tetrahedral Occupancy Cells

Another common approach is to instead use binary sub-division and geometric simplexes to partition the data. In 3D the simplexes are tetrahedra, geometric primitive polyhedra, consisting of 4 non-coplanar vertices and 4 triangular faces (Figure 4-11). Tetrahedra are natural objects for 3D geometry. They are defined by their faces, triangles which current graphics card hardware use as a basis for visualization and acceleration of numerical computations. The ideas for tetrahedral occupancy cells are similar to the cube-based hierarchical structures, except the initial cube is first partitioned into 6 tetrahedra which will be the base nodes (see Figure 4-11). Each of these tetrahedra has the main diagonal of the cube as its longest edge. As described below, further sub-division of each node (i.e., tetrahedra) is performed by splitting the longest edge of the tetrahedral which results in two children tetrahedra. The octree structure from hierarchical cubes is then replaced by a binary tree structure of tetrahedra. The surface boundary of the union of the occupied tetrahedra at any stage of sub-division can be thought of as a rough, 0-th order approximation of the surface of the point cloud. For higher order approximation, the different tetrahedra must be organized and linked together in case some sort of stitching is necessary.

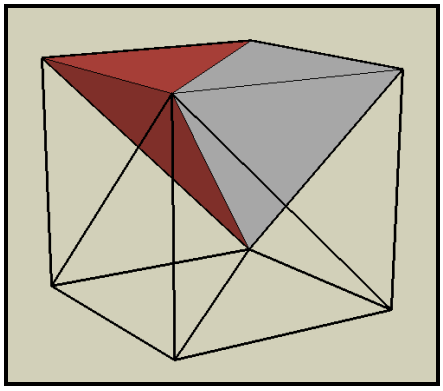


Figure 4-11: Cube Partitioned into Six Tetrahedral Simplices.

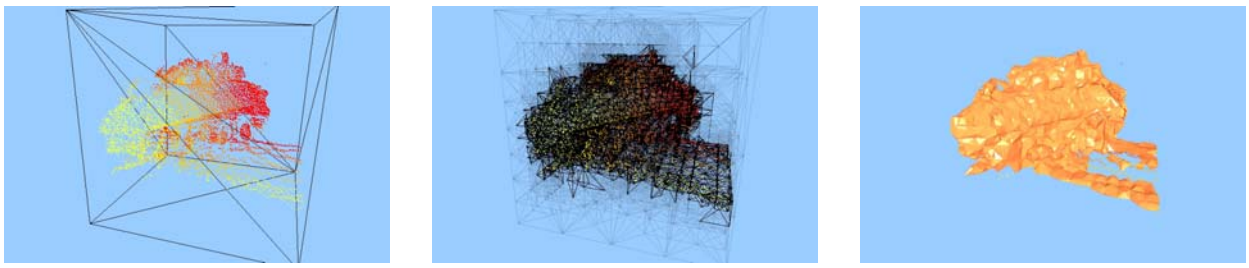
This will require additional structure since trees do not guarantee that nodes that are spatially close are necessarily close in traversing the tree. Moreover, when using adaptive sub-division schemes, discontinuities along the boundary of the nodes in the form of hanging vertices are introduced and must be handled. For example, when using a marching cubes scheme on a multi-resolution octree, one must take special care to ensure that the surface produced is continuous between neighbouring cubes that potentially differ greatly in size or by their appearance in the depth of the octree. Note that using this scheme on all six tetrahedra derived from the cube, the levels of an octree sub-division emerge from every three uniform sub-divisions of the tetrahedra trees. That is, if both the octree scheme and the tetrahedra scheme start from the same cube, the same set of vertices will be used by the leaf nodes in both schemes.

At this point the tree of tetrahedra has a frontier, or set of leaf nodes, which can still fall victim to the same discontinuity problems mentioned previously. To prevent this, a conforming step is performed by sub-

dividing additional tetrahedra while building the tree. To sub-divide a single tetrahedron the longest edge is selected and a new vertex is added at its midpoint. This new vertex is then connected to the two vertices not on the longest edge giving two children tetrahedra. As the new vertex generally introduces a discontinuity, i.e., a node in the middle of an edge of a neighbouring tetrahedron which shared the original edge, we additionally sub-divide all other tetrahedra that share this split edge. The additional sub-divisions result in a smooth transition from the finer regions of the frontier to the coarser regions. This allows algorithms like *marching tetrahedra* to be used without concern for special cases along boundaries. Also, since this scheme is simply a collection of binary trees, connections between nodes to allow for fast traversal of nodes at any level in the tree can be added, including links for traversal of the varying levels of the frontier, or leaf, nodes. Thus nodes that are spatially close are ensured to be close in the tree structure.

With the additional links between the nodes of the tree, locally stored statistics in the structure of each node can easily be distributed for analysis to neighbouring nodes without global searches. Moreover, if new data arrives, then the local information can be easily updated, and splitting criteria reapplied for possible finer resolution based upon the new data. Particular statistics of interest would be the barycentre and covariance matrix of that portion of the point cloud within the cell, since they indicate spatially and structurally similar data. The statistics or data stored in each node are application dependent and do not necessarily have to be related to the splitting criteria. With this information the splitting and stopping criteria can be modified to create trees that provide varying types and amounts of information. For example, if only occupancy information is of interest, a simple threshold on the number of points per node may be used. If instead the space should be sub-divided until each node contains roughly planes or lines, then PCA can be performed on a node individually or a collection of neighbours. Also, local curvature information can be computed and easily compared to neighbouring nodes or even coarser or finer nodes.

As mentioned earlier in this section, multi-resolution tetrahedra or cubes could be used as the support for local piecewise polynomial surface fits of the point cloud by using methods from Computer Aided Geometric Design (CAGD). This is in fact being implemented and tested by Texas A&M University and Rice University personnel of the ARO MURI in the Wavelet Streaming Surface Reconstruction (WSSR) algorithm [102]. For the purposes of this report we focus on how to integrate techniques from the next section on implicit surfaces for hybrid versions of point cloud assimilation.



a) **b)** **c)**

Figure 4-12: a) Point Cloud of Wright-Patterson ASC Flash LIDAR Data of DRDC Valcartier Armoured Personnel Carrier Rendered by Range Colouring; b) Partitioned into Occupied and Force-Split Tetrahedra; and c) Rough Surface Representation Formed by Surfaces of Leaf Node Tetrahedra.

4.2.1.3 Comparison of Oct-Cubes and Tetrahedral Bin-Trees

Before proceeding to the discussion of implicit surfaces, a brief comparison is provided for octree-cube and binary tree – tetrahedron occupancy structures. As is commonly the case, there is a play off in each case between progressively storing information into structures for quickly determining neighbours (memory drain) and re-computation of local properties to traverse the tree structures.

Oct-cubes use octrees for their structures, but can utilize very simple indexing for parent/child and spatial neighbours, which results in minimal overhead in terms of memory for pointers and structures and faster implementation. On the negative side, oct-cubes are non-conforming and are less reliable when integrated in implicit surface algorithms which produce level sets and surfaces.

Bin-trees used for tetrahedra are simple to encode, but the gain in speed for processing operations require additional overhead in memory (in addition to pointers for parents and children, pointers also used for face neighbours and vertex neighbours for leaf nodes). Implementation effort is also substantially more involved. However, multi-level tetrahedra are more efficient for graphics hardware computations and for more efficiently capturing a point cloud in 3D. As such the local statistics for tetrahedra should on average be more reliable.

4.2.2 Implicit Surface Representations

In an implicit representation in 3D, the surface is described as the set of points which are related implicitly through their relationship in satisfying an equation of the form:

$$F(x,y,z)=0\,.$$

Thus, the surface can be viewed as the boundary of the level set of a function F defined on a domain in R^3. There are many functions F which yield the same surface. One particularly useful choice is the function $\phi = F_d$ which gives the *signed* distance of the point (x,y,z) to the surface. The sign will be positive if the point is exterior to the surface and negative otherwise. This choice is particularly useful for improved approximation of surface fits of finely distributed point clouds, and in line of sight calculations and autonomous navigation.

A point (x,y,z) is in the *line of sight* from (x_0,y_0,z_0) if the distance function ϕ does not change sign on the line segment connecting these two points. In autonomous navigation, a fly zone can correspond to the buffer region $\phi \geq \delta$ where $\delta > 0$ is some prescribed tolerance based upon the vehicle's control and operational characteristics.

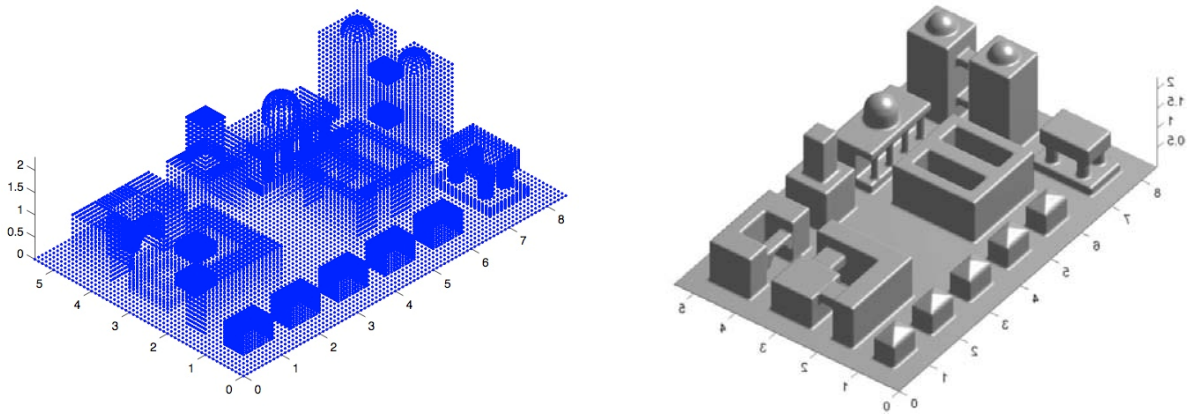
The most essential step in an implicit representation of a surface is a separation and classification of the whole space domain into two regions: interior and exterior. The boundary between these two regions is the surface. For representation, compression and reconstruction, one only needs to know and store information near the boundary which fits the point cloud. Of course the interior region can and will generally, be multiply connected, e.g., many buildings separate from each other, all defined simply through one scalar function.

Other possibilities than the distance function are possible for the function F to provide an implicit fit to a point cloud or which would have a prescribed set as a surface interior. Variational problems can be formulated in which a sensed point cloud acts as a constrained attractor to construct a surface fit to unorganized data. ϕ can be thought to act as a potential function for the point cloud. The objective is to find a local minimiser of an energy functional that behaves like a minimal surface or an elastic membrane variably attracted to the data set. The potential force and the surface tension are balanced in a variational equation where the solution ϕ must satisfy a corresponding Euler-Lagrange equation [49] of the form:

$$\frac{\partial\phi}{\partial t}=|\nabla\phi|\left[\nabla d(x)\cdot\frac{\nabla\phi}{|\nabla\phi|}+\frac{1}{p}d(x)\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\right].$$

Here, the first term in brackets corresponds to the attraction by the distance field d and the second term corresponds to a minimal surface regularization weighted by the distance function d to the point cloud. Recall that this last term consisting of the divergence of the unit normal of ϕ is just the curvature.

Implicit models are a natural choice for the modelling of 3D urban terrain because of their flexibility and robustness in dealing with complicated topology. Every object can be regarded as solid or volumetric. Hence one can mark regions that are inside structures or underground as interior. The whole complementary region is marked as exterior. Line-of-sight information can be used to distinguish interior and exterior regions. Non-genus-0 topology poses no extra difficulty. Other information, such as connectedness, can also be incorporated to compute the correct topology [50]. Usually the interior region is composed of multiple disconnected components which are associated with the interior of different objects. Many operations and manipulations become very simple using implicit methods, for example, Boolean operations, finding intersections, visibility and path planning [51],[52] – see Figure 4-13.



a) b)

Figure 4-13: a) Synthetic Point Cloud of a Simulated Urban Terrain with Moderately Complex Topology; and b) Level Set Reconstruction of Sensed Surface Using Variational Surface Fitting.

The advantages of implicit representation in image processing are now well demonstrated. They lead to state-of-the-art denoising and deblurring algorithms [53]. There are several compelling reasons to utilize implicit representations in terrain processing. The first is captured in the remarks above on line of sight and navigation. The second has to do with efficient encoding. Some surfaces or portions of surfaces are much easier to describe in implicit form. This is the case for most man-made structures. Consider for example the cylindrical surface $x^2+y^2 = r^2$, $0 \leq z \leq L$, which could correspond to a portion of a telephone pole or light pole. Other man-made structures often have similar simple representations. In implicit form this surface can be described by very few bits. Its description by explicit methods (for example approximation of its level curves by piecewise linear functions) would be much more costly. Another point is that implicit representations more easily describe the topology of the surface and connectivity regions.

4.2.3 Hybrid Methods

There is no simple answer to the question as to which of explicit or implicit methods are superior in general. Hierarchical occupancy cells and the analysis stages of explicit methods are very fast and scale linearly with the data. Implicit methods have advantages of fast line of sight, high quality denoising and deblurring, but typically require dense data and computationally expensive processing. Thus, an ideal framework would be an integrated system which has the benefits of both: the potential to utilize the speed and localized analysis of explicit methods while exploiting level set properties of implicit methods. This is the main focus of the ARO funded Multi-University Research Initiative entitled “Dynamic Modeling of 3D Urban Terrain” [49],[53],[101],[102]. One general resource for rural and urban terrains is maintained at the Virtual Terrain website [103]. There are detailed links which provide background for usage of terrain and cultural models; ontologies; data sources and formats; methodologies for data acquisition and fusion, classification and geometry building; and references for commercial software.

4.3 CLASSIFICATION OF TERRAIN OBJECTS

Many applications, such as simulations, require explicit information about the types of the individual objects in the scene. Clearly, the sensor readings themselves do not carry any such high-level information but it has to be added afterwards in a *classification* step, either manually or – as will be the topic here – through automatic data analysis.

From a modeling point of view, the classification functionality is an important component, as it allows for creating application-specific model instantiations, in which each classified segment is replaced with a 3D model component. This means that whence we have classified a particular segment into a certain class, say, a tree, we can replace the original data points stemming from the tree with a 3D model of a tree, chosen so as to be relevant in the application at hand.

4.3.1 Bare-Earth Extraction

The ability to estimate the height of the ground, or bare-earth, surface is typically of fundamental importance for creating an explicit 3D model of a particular scene. Such a ground model, often referred to as a *Digital Terrain Model*, or DTM, is the base on which other model components – building, trees, cars, lamp posts, trash bins, etc. – are placed in order to form the final 3D model. From a signal processing point of view, having access to a DTM facilitates the scene analysis significantly as it limits the space in which to look for, and expect, certain kinds of objects. In addition, the DTM in itself is often used in many geotechnical applications, e.g., to assess risks of flooding, landslides, etc. An example of a DTM created with a flexible surface matching technique is presented in [55]. In principle, all existing DTM estimation techniques make use of one fundamental assumption: *ground data points have low elevation values*. Hence, the first step in ground modeling is to establish a set of very probable ground points based on the expected nature of the ground in the particular region, see Figure 4-14. Then, based on an initial set of ground points, the ground level at other locations in the region can be estimated through either successively classifying more points as ground and/or interpolating (or extrapolating) the ground level in regions void of classified ground points. Overviews of different DTM estimation techniques can be found in [56],[57].

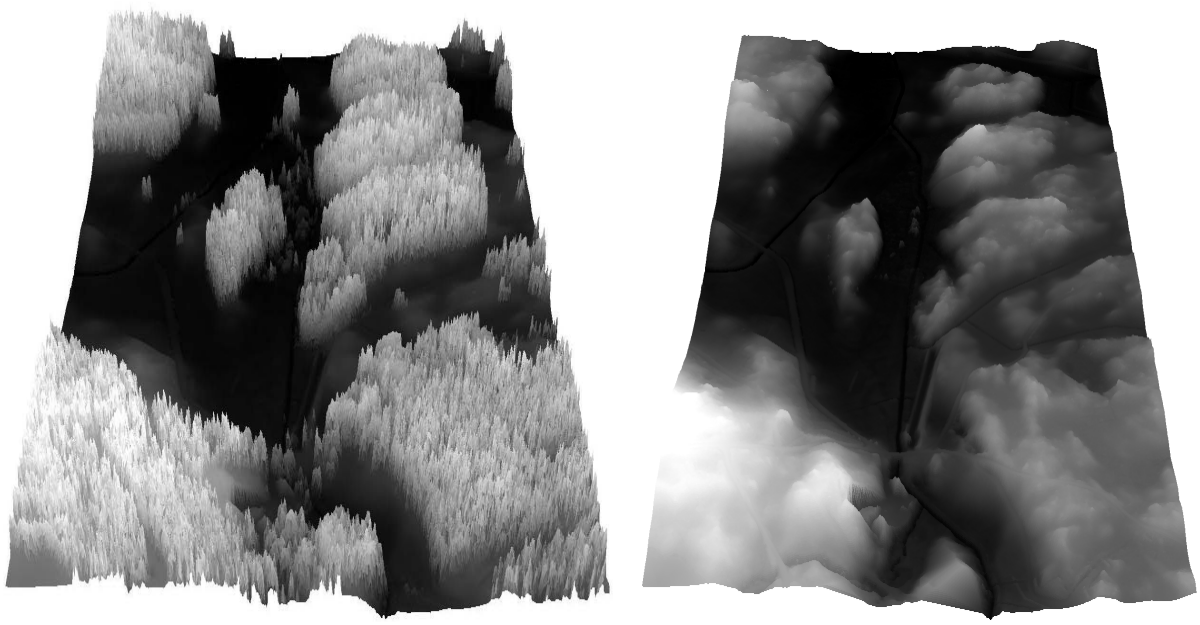


Figure 4-14: (Left) Elevation Data (Digital Surface Model); and (Right) Ground Surface (Digital Terrain Model).

4.3.2 Classification of Above-Ground Objects

We now turn to the task of classifying the objects that are located upon the ground, here referred to as *above-ground objects*. Analogous to the problem of distinguishing between ground and non-ground data points, we are seeking a set of *characteristics* or *features* that can be used to separate the classes. In order for the classification to succeed, the features for a particular class should be different from other classes and not vary too much within the class. Put differently, good features should display *high inter-class variation* and *low intra-class variation*. What features can or should be used depend on a lot of circumstances, e.g., what types of objects are considered, what accuracy/detail level is required and what sensor data are available. In the literature, many different features have been proposed and it is beyond the scope of this report to go into details concerning specifics of their nature [57]. Nevertheless, the features are designed so as to capture certain relevant underlying physical properties of the objects, and the variation among those properties is more tractable than that among the plethora of suggested specific features and algorithms.

Here we will focus on two especially common ingredients in 3D urban models – buildings and trees – and give some examples of physical properties that are exploited for classifying these objects. In practice, several properties are often combined to achieve robustness to changes in appearance of objects within the same class. With continuously increasing resolution and accuracy of airborne range/3D measurement systems, it is likely that the future will behold more efforts on automatic recognition of smaller and finer objects, such as fire posts, wells, walls, antennas, pedestrians, vehicles, etc.

4.3.2.1 Buildings

- **Relatively Large –** Many (most) buildings are large compared to other objects expected to appear in the scene. This means that a classifier using only a simple size feature would probably do a good job on classifying the major buildings in the scene.

- **Regularly Shaped Footprints –** Most buildings have a regular shape, in the sense that they consist of (linear) wall segments that are oriented in a structured fashion. However, not all regular buildings appear as regular structures in the data. This can be due to occlusion effects (e.g., trees hanging over the building, thus altering its apparent shape) or insufficient data density (causing walls to appear as jagged). Obviously, the smaller the building and the heavier the occlusion, the more difficult it is to use the edge information for robust classification. In fact, at some point, it becomes difficult to distinguish the footprint of a building from that of tree.

- **Smooth Roof Segments –** We expect the geometry of roofs to be smoother than that of vegetation. In fact, building roofs can often be successfully described as with a set of geometrical primitives, e.g., flat surfaces. Obviously other roof shapes exist, such as conical or dome-shaped, that can be treated following the same line of principle. From a signal processing point of view, the smoothness/flatness can be quantified by computing the residual of fitting a plane to a set of points, by estimating local height variations, by local statistics (e.g., PCA, covariance), etc.

- **Homogeneous Objects –** A LIDAR sensor can make use of the fact that buildings are typically homogeneous objects. Among other things, this implies that it is impenetrable for the laser light and hence that we do not expect to find hits from the ground within the building, the exceptions being skylights or courtyards. This property can also be used for segmentation of laser data into homogeneous regions (see Figure 4-15). Note the problems that occur at the border of the buildings, as the roof typically extrudes from the wall. Nevertheless, the interior of the building is typically void of ground hits.

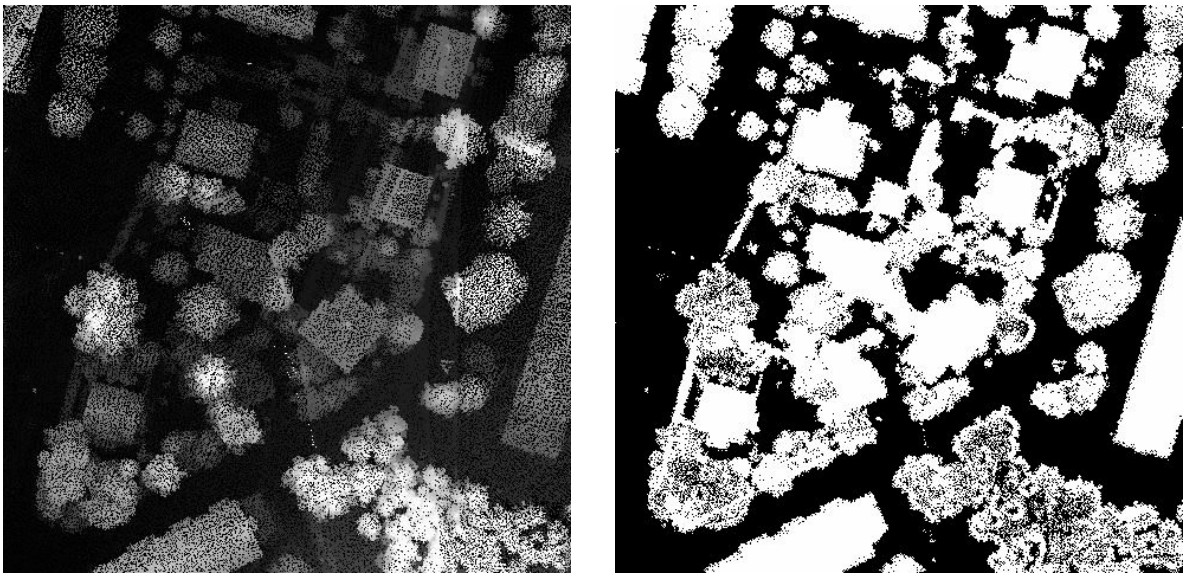


Figure 4-15: (Left) Elevation Image; and (Right) Black Pixels Indicate where Ground Hits have Occurred. Note the difference between the buildings and the trees in this respect.

4.3.2.2 Trees

- **Inhomogeneous Objects (many small reflecting surfaces at different depths)** – Trees are often only partially obscuring the ground beneath them, and as a consequence there are often laser hits from both the tree canopy and the ground beneath, see Figure 4-15. In addition, the vegetation areas display seemingly random height variations which can be quantified through Principal Components Analysis, a height variance filter, etc. As mentioned earlier in this report, some modern LIDAR systems provide the possibility of collecting the entire received waveform. This waveform is the superposition of all the reflections back to the detector within the laser beam footprint. Since trees consist of many small reflecting surfaces at different distances from the laser system, pulses reflected off trees generally broaden considerably. This could prove very useful for distinguishing between trees and hard surfaces [57].

- **Local Height Maxima** – If sampled densely enough, trees typically exhibit the shape of local height hills, especially after low-pass filtering of the height data. Finding such hills makes it possible to detect single trees. In passive imagery, the height hills may cast shadows on their neighborhood that can be detected to support the detection of the height hills in the spatial domain.

- **Spectral Characteristics Often Different from Those of Buildings** – Since vegetation and building roofs typically consists of different materials, their spectral appearance is expected to differ. For example, trees with leaves often display a prominent green color whereas buildings roofs do not. The difference is often obvious within the RGB domain, but may become emphasized if more spectral bands are used, e.g., NIR (near infra-red). Hyperspectral data have become commercially available, which further increases the class-separating potential of the spectral data. Obviously, the success of classification based on spectral properties hinges on managing spectral variations caused by seasonal changes (leaves vs. no leaves, snow, etc.), shadows, sun irradiation, angular-dependent reflectance functions, sensor noise, etc. Ideally, if one could compensate for all such effects, the result would be a spectral response curve that could be matched to a spectral library. In practice, however, the most efficient way of classifying spectral information data is often to train a classifier on a representative portion of the data at hand and have it learn the spectral characteristics of the region under study.

4.3.3 Classifier Examples

At FOI the problem of automatic detection of buildings and trees has been studied for the last decade. As a result, three individual classifiers have been developed that are all based on different techniques for segmentation and classification, each with its own merits and disadvantages. One technique [58],[59] segments the data using multiple echo information and classifies the resulting segments using shape and height variation features, including Hough transform-based features for recognizing regular building footprints. Another technique [59] aims at first segmenting the data into regions containing no interior ground hits, using the fact that laser beams often penetrate canopies and hits the ground below, while being stopped effectively by most roof materials. Large segments that contain flat surfaces (i.e., roofs) are then marked as potential buildings. A third technique [60] performs a connected component (region growing) segmentation of the data and relies on Principal Component Analysis to distinguish planar segments (roofs) from other types of objects.

In order to get a more detailed understanding of the performance, a limited region was selected and analyzed thoroughly, object by object. Table 4-1 contains an example of a summary of such an analysis for one particular building detection result (basically all techniques displayed similar results). The purpose of this misclassification analysis was to learn more about the cause of the errors produced by the algorithm with respect to ground truth (here, a city cadastral map). After all, there are many elements that affect the overall performance – from data acquisition, platform positioning and alignment of swaths to segmentation and classification. Among other things, the results indicated that many of the undetected buildings were (at least partially) occluded, and hence that those objects had not been accurately measured with the LIDAR system. Furthermore, other buildings had only very few hits on the roof, as of consequence of roof material that caused data dropouts due to very unfavorable reflectance properties, while other buildings were simply too small so that no structures could be detected reliably from the sparse returns. A large portion of the erroneously detected objects had a regular geometry that made them look like buildings, only they were not actually buildings according to the cadastral map, but containers, tents, etc. In short, this analysis indicates that the dataset itself is a main limiting factor here.

Table 4-1: Detailed Analysis of Erroneously Classified Objects, as Compared to Ground Truth.

False Negatives	#	%
Number of Undetected Buildings	**40**	
Occluded building	14	35.0%
Very few LIDAR samples	8	20.0%
Small/low building	10	25.0%
Building not present in LIDAR data	3	7.5%
Only part of building present within image	5	12.5%
False Positives		
Number of Erroneously Detected Buildings	**50**	
Natural objects (trees, bushes, water, etc.)	20	40.0%
Human-made, regular objects (vehicles, containers, etc.)	30	60.0%

4.3.4 Data and Information Fusion

As mentioned above, the quality of the data itself, for example in terms of sampling density, range noise, occlusion, etc., is an important limiting factor. Thus, what would really increase overall performance is richer data that carry more information about the objects of the urban terrain. As richer sensor data is starting to appear frequently on the market, e.g., LIDAR waveform, simultaneous spatial and spectral measurements,

geolocation, fusion of terrestrial and airborne data, structure-from-motion data, hyperspectral data, SAR imagery, etc., the future will probably hold an increased amount of integration of different kinds of data, at the expense of reduced development efforts concerning "LIDAR only"-based mapping techniques.

In practice, classification of objects in urban terrain based on geometrical properties generally enables quite good performance, provided that the original data are dense and accurate enough to capture fine details of the sought-after objects. However, there will most often, if not always, be flaws in the data due to occlusion, range noise, poor resolution, dropouts, etc., that will lead to erroneous decisions based on the particular data. In order to improve the chances of making correct classifications, various information cues can be used and combined, referred to as *fusion*. Having access to raw data enables us to design classifiers that combine data through *data*, or *signal-level, fusion*, e.g., LIDAR data and aerial imagery that allows to exploit both spatial and spectral features in the classification process.

With a rapidly increasing number of commercial software and services for automatic analysis of sensor data, it is likely that the end-users will not always have the sensor data at their disposal, let alone have enough knowledge of how to design robust features and classifiers. In addition, the users may have complementary information such as already existing maps, old classification results, etc. In such cases, the user would benefit from fusing data on a higher level, e.g., through majority voting among several classifiers, which is an example of *decision-level fusion*. In Table 4-2 the results of an information fusion (through majority voting) are presented.

Table 4-2: The Table Shows the Results of Information (Decision-Level) Fusion of Individual Results Through Majority Voting. Intentionally, none of the techniques were optimized for the current dataset. This is most notable for the techniques referred to as "2003" (high FAR) and "2006" (low P_d).

Voting Methods \ Data Test Sets	2003	2006	2007	Fusion
P_d	0.830	0.701	0.811	0.801
Number of detected known objects [km^{-2}]	205	173	199	197
FAR [km^{-2}]	168	44	65	54

4.3.5 Building Reconstruction from Airborne LIDAR Data

Many (most) 3D modeling or visualization software packages require that 3D objects are represented in a vector format, such as CityGML, kml, 3ds, OpenFlight, etc., rather than by a collection of 3D points or an implicit model.

Basically, the modeling can be performed at an arbitrary level of detail, from very simple and coarse "shoe box" models to highly detailed 3D models capturing every significant geometrical feature of the building. If the sampling density is high compared to the geometry variations within the building, which is often the case with modern high-resolution sensor systems, it is often possible to apply automatic algorithms for creating a realistic 3D model for each individual building.

The general approach to the polygon modelling problem is to fit a set of predetermined geometrical primitives to the data and by doing so determining the primitives and the topology that best describe the building. This is how most existing building modelling techniques work, although they may differ with respect to the specific techniques used to determine the primitives and the post-processing applied. By far the most common geometrical primitives are planar surfaces and line segments. See Figure 4-16 for an illustration of some key steps in the process of creating a 3D model from elevation data.

One can distinguish between two main classes of building modelling approaches: *bottom-up* and *top-down*. Techniques in the bottom-up class start by gradually extracting local primitives from the data that are eventually combined into a 3D model. Such techniques are inherently capable of modelling any

building that can be described with the set of primitives. In practice however, the performance is often limited by the quality of the data – sampling density with respect to the spatial variation of the building roof, noise, occlusion, etc. In addition, a good 3D modelling result obviously requires that the geometrical primitives (often planar surfaces, ridges, break lines, etc.) match the real characteristics of the building. Trying to model a cylindrical building using only piecewise linear wall segments is clearly not ideal but could still result in a decent-looking and useful model, depending on the requirements of the application at hand.

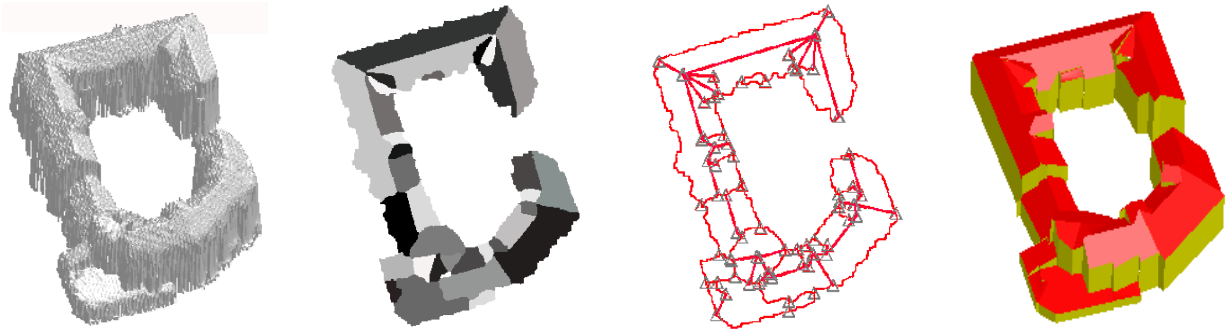


Figure 4-16: Illustration of the Process of Creating a 3D Building Model from Airborne LIDAR Data. From left to right: original elevation data belonging to the building; detected geometrical primitives (here, planar patches); detected keypoints and connectivity lines extracted from topology analysis; and the final 3D model.

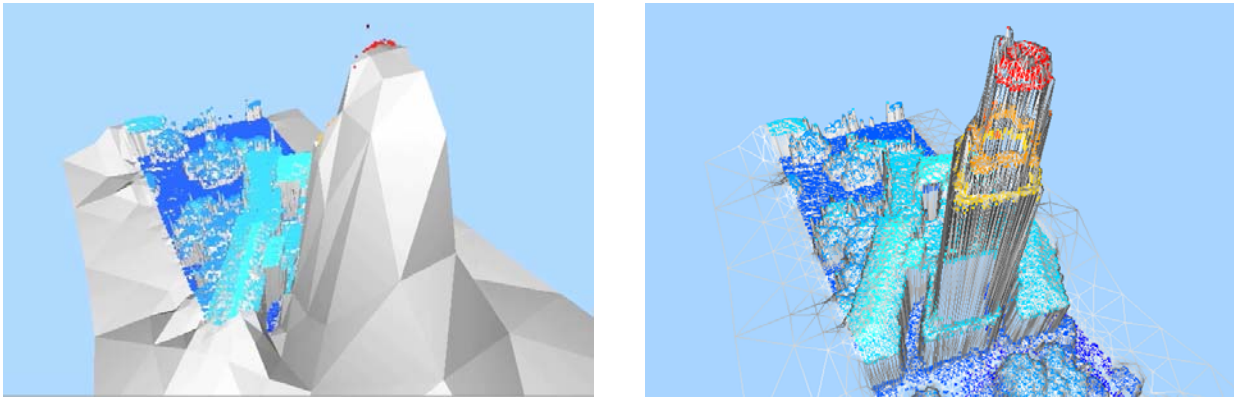
The core in the top-down approach is hypothesis testing – a pre-defined set of prototype building models, possibly with some tuneable parameters, is fitted to the data and the one yielding the highest score is selected as the winner. The use of prototypes in the matching process ensures nice-looking results but restricts the variations in the final building 3D models to the diversity of the prototype library. In addition, the computational complexity increases with the number of prototypes.

4.4 SAMPLE PROCESSED SET-118 DATA SETS

NATO SET-118 members have contributed collections of point clouds as data for use in testing various processing methods such as ability to handle non-uniformly spaced data, significant noise levels, to distinguish occlusions from missing data, and to accommodate complex geometry and topology.

4.4.1 WP-AFRL: PILAR (Polarimetric Imaging Laser RADAR)

2.5D LIDAR data was provided by Wright Patterson personnel (US AFRL – RY JM) from their line scanning airborne system [97]. The data was processed using South Carolina's on-line adaptive partitioning algorithm [98],[4], while progressively updating the surface as data is acquired. This is illustrated in Figure 4-17. In part a) data (point cloud coloured by height) has not yet been received in the foreground, but the gridding adjusts as data is acquired in order to better reflect the geometry. In this case the surface is a functional surface with simple topology. The final surface is shown in part b).



a) b)

Figure 4-17: a) Gridding of PILAR Data Swath is Adaptively Refined in Order to Estimate Heights of Local Surface Patch to Fit the Point Cloud; and b) Surface Statistics are Progressively Updated On-Line as Data is Received.

4.4.2 WP-AFRL: Flash LIDAR Data

Flash LIDAR data (see Section 3.2.1 for a detailed description) was provided by Wright Patterson (US AFRL – RY JM) personnel as part of SET-118's Bonnland data acquisition effort in November 2007. This data set was collected from a ground vehicle and is heterogeneously rich with some areas providing highly detailed point clouds (Figure 4-18), while others exhibit many of the problems common to imaging in urban settings, such as distinguishing missing returns, entrances, and occlusions in analyzing the data (Figure 4-20).

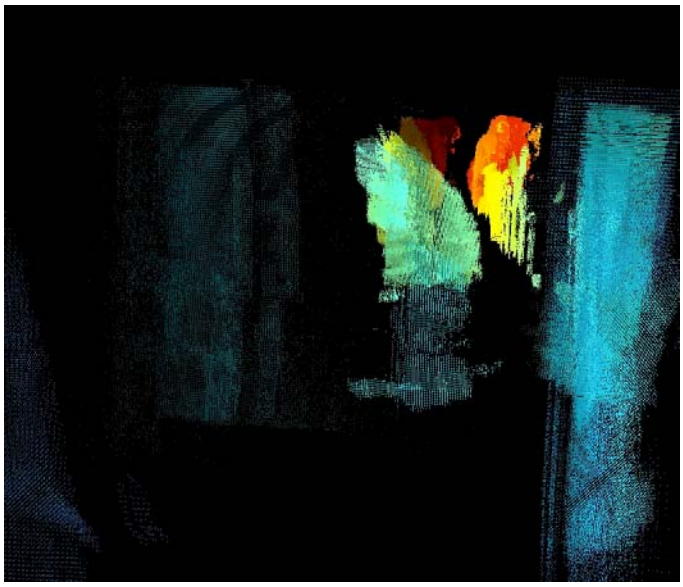
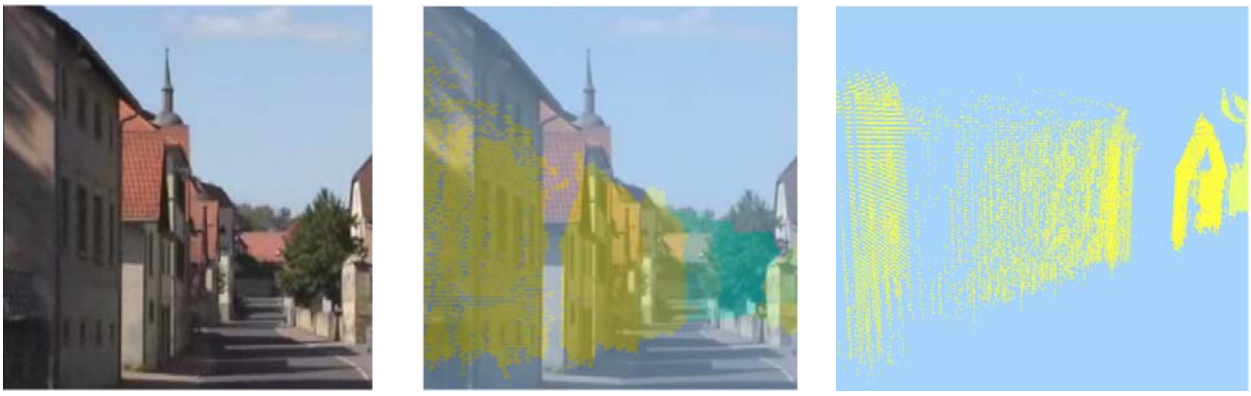


Figure 4-18: Rendering of Post-Assembled Bonnland Point Cloud, Time-Coded by Colour from Blue to Red, from a View Looking up the Roadway Haupstrasse.

In Figure 4-19, we have overlaid upon a frame, taken from the Bonnland video at the time of collection, the point cloud from an approximate viewpoint as the flash LIDAR in order to provide rough reference. Figure 4-20 illustrates natural problems which arise in surface reconstruction. Figure 4-20a) is a zoomed-out

view the full point cloud collection, with colour rendering used to denote acquisition time (red to blue) of the flash LIDAR images, which were later registered using an ICP algorithm. The LIDAR sensor was pointing forward as the platform moved toward the Northeast up the roadway of Haupstrasse. Figure 4-20b) is a rendering of two buildings along the right-hand side of the acquisition pathway. Figure 4-20c) is the result of surface reconstruction of the point clouds of the two buildings using a multi-resolution distance field. The holes in the point clouds (and therefore the reconstructed surfaces) are the result of both occlusions from the push-broom effect of a fixed sensor orientation on the platform as it moves forward from left to right and of data sparsity. In this particular case, the holes in the left hand sides of the surfaces cannot be distinguished between an entrance way, missing data, or occlusion without knowing the exact sensor state (position and pose) at the instant of acquisition. In the absence of further assumptions on the data or access to metadata information, such as time coding of the sensor state, inherent ambiguities such as distinguishing between occlusions and missing data will be unresolved.



a) b) c)

Figure 4-19: Referencing the LIDAR Point Cloud to Video from the Sensor Platform – a) Photo extracted from video; b) Portion of point cloud overlaying photo; and c) Rendering of that portion of the point cloud imaged from the first two building in the left foreground.

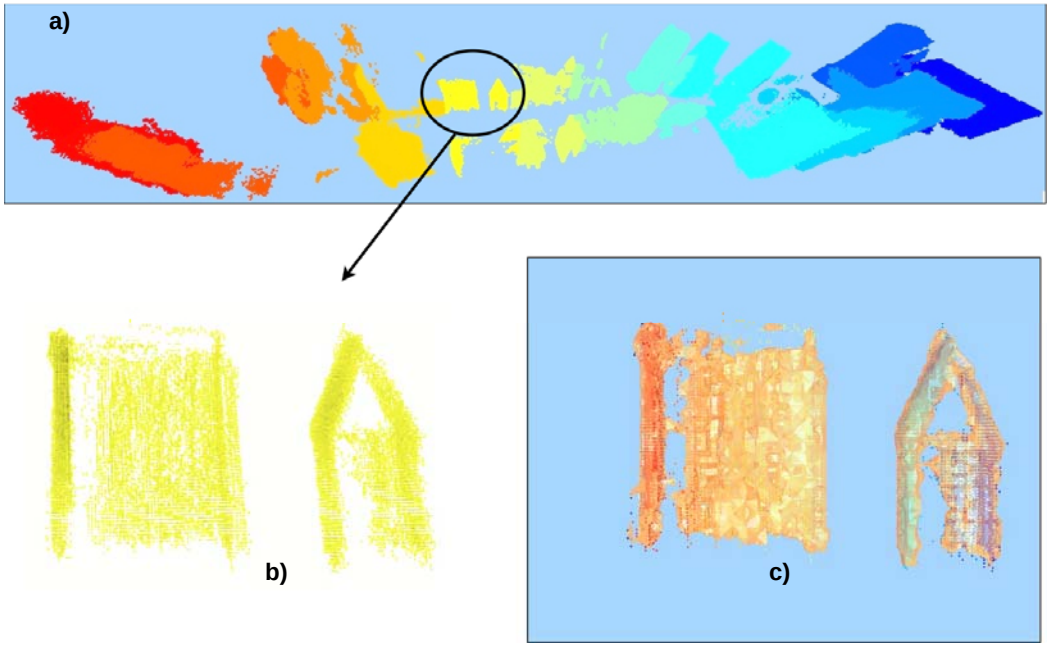
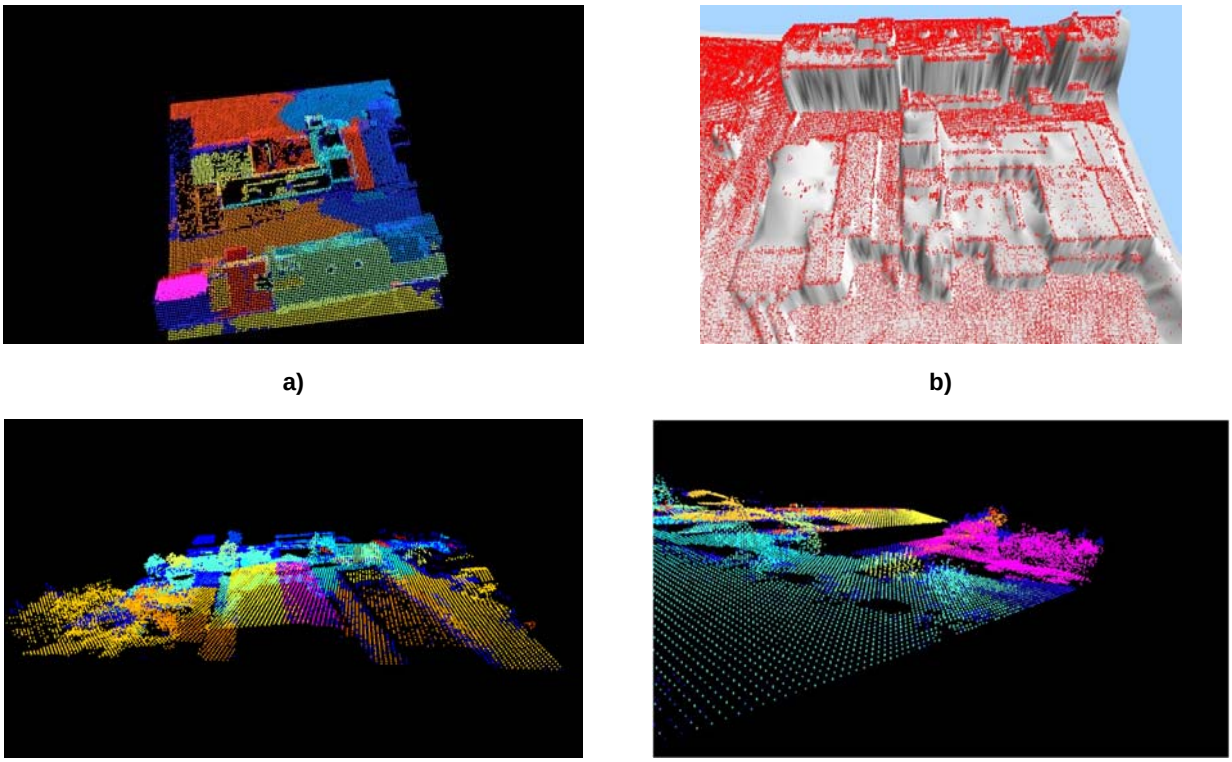


Figure 4-20: Bonnland Data Features and Processing – a) Zoomed out view of full extent of point cloud collection, again with colour rendering used to denote acquisition time of the flash LIDAR images, which were later registered using an ICP algorithm; b) Zoomed-in view of two buildings along the roadway; c) Surface reconstruction of the two buildings using a multi-resolution distance field.

4.4.3 FOI's Norrköping Collection

SET-118 members from Sweden's FOI and University of South Carolina in the U.S. have collaborated on testing promising segmentation and surface reconstruction ideas which have been developed in an ARO supported MURI. This data was fully described in the previous sub-section (Section 4.3), along with a complete description of the analysis and subsequent classification of urban objects, such as bare earth, waterways, and above ground objects such as various types of buildings, vehicles, roads, paths, power lines, trees, and other vegetation. In this brief sub-section, only the automatic processing of the Norrköping data by SET-118 participants is addressed, in order to test the effectiveness of automatic segmentation and reconstruction algorithms.

Two regions were cropped from the Norrköping data set. The point cloud for the cropped region #1 is shown in Figure 4-21a) where the *iterated Mahalanobis* algorithm [99] has been used to 'cluster' points and classify them by a colour map. The algorithm is based on a multi-resolution version of statistical Principal Components and used to determine local coordinates and corresponding distances in order to identify outliers to a given segment. The algorithm iteratively grows the current segment capturing all points in the cloud which are judged to be close in this distance. The algorithm may be implemented with computational complexity $O(N \log N)$. Figure 4-21b) is a surface reconstruction of using adaptive partitioning and mathematical learning of non-parametric point distributions [98]. Figure 4-21c)-d) are given to illustrate the limits of this algorithm for cropped region #2 in the absence of additional meta information. In Figure 4-21c) the points in the point cloud are again classified by colour, according to the local Mahalanobis distance. The magenta coloured set of a sloped roof is well defined from its neighbouring segments. In Figure 4-21d) however, the magenta coloured collection of points is not able to separate a roof-top from neighbouring trees. Principal Component analysis, by comparing eigenvalues, is able to individually distinguish trees (as was the case in Figure 4-21c)), so it is not clear if an adjustment of the algorithm, especially in its multi-resolution instantiation, will be able to execute the separation.



a) b)

c) d)

Figure 4-21: Segmentation and Surface Reconstruction of Cropped Portions of Norrköping Data Set – a) Point cloud for cropped portion #1 which is segmented using the iterated Mahalanobis algorithm with different segments classified by colour; b) Surface reconstruction using multi-resolution learning algorithm for surface heights for point cloud – after a rotation of about 180°; c) Point cloud for cropped region #2 with sample segment – a slanted roof – rendered in magenta; and d) Point cloud for cropped region #2 demonstrating the difficulty of segmenting merged features, in this case a building roof and adjoining trees (rendered in magenta).

Figure 4-22 shows an example of automatic terrain classification for the Norrköping data. The classes are ground (black), buildings (pink/red), vegetation (green), unknown (yellow) and water (blue). The classification is based on geometrical features except for the water class which relies on the fact that water regions manifest themselves as large regions void of laser hits.

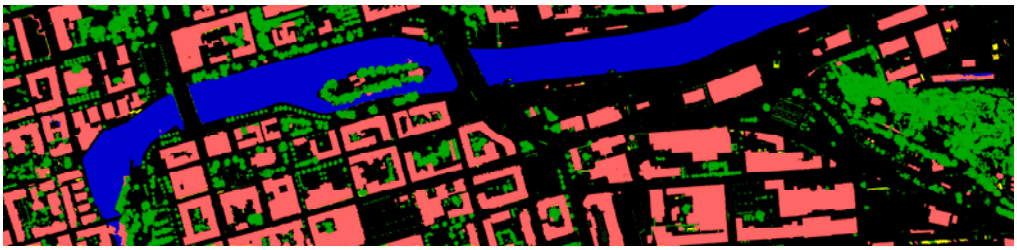


Figure 4-22: Results from Automatic Object Classification for a Part of Norrköping.

In Figure 4-23, some of the 3D building models for the city of Norrköping have been put on the bare-earth model and colorized using an aerial photo. Prior to the 3D modelling of the buildings, they were detected using algorithms discussed in Section 4.3.3.

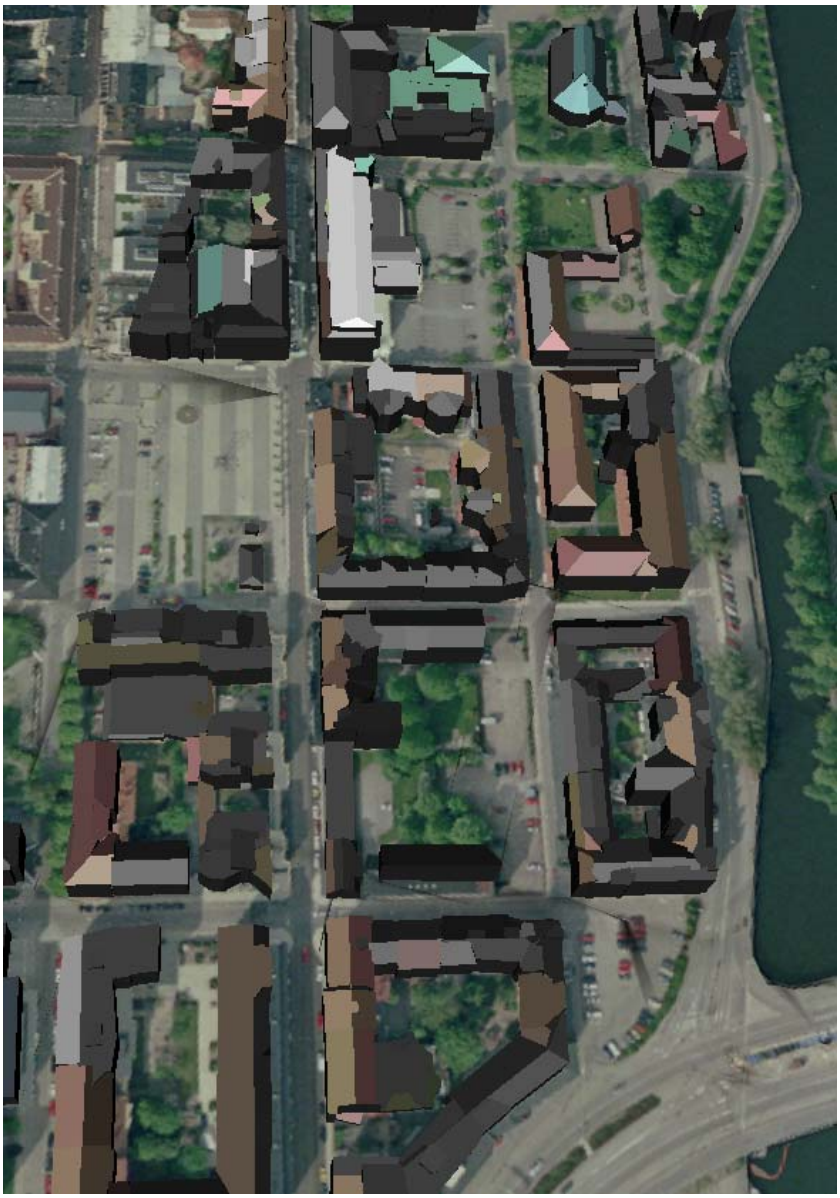


Figure 4-23: 3D Building Reconstruction of a Cropped Portion of Norrköping Data.

4.4.4 UK – Defence Science and Technology Laboratory's Burst Illumination LIDAR

The original objective of the Mahalanobis distance formulation was to determine distances between normally distributed multi-variate distributions, and to determine outliers for each. This feature of the iterated Mahalanobis algorithm is consequently very effective for denoising raw sensor data as a pre-processing stage prior to geometric reconstruction. Figure 4-24a)-d) present the results of such an algorithm [99] applied to burst illumination LIDAR data provided by NATO SET-118 collaborators from the Defence Science and Technology Laboratory. Figure 4-24a) is a photo of the imaged scene, with parts b)-c) representing side and front views of the sign in the foreground. In these images, magenta is used to colour the residual set after the iterated segmentation. Figure 4-24d) is a rendering of the surfaces constructed from the segmented sub-clouds along with the original data. Figure 4-24e)-f) show similar side and front views for the vehicle in the background of the photo in part a). The side views of the target board and vehicle indicate the potential of the method.

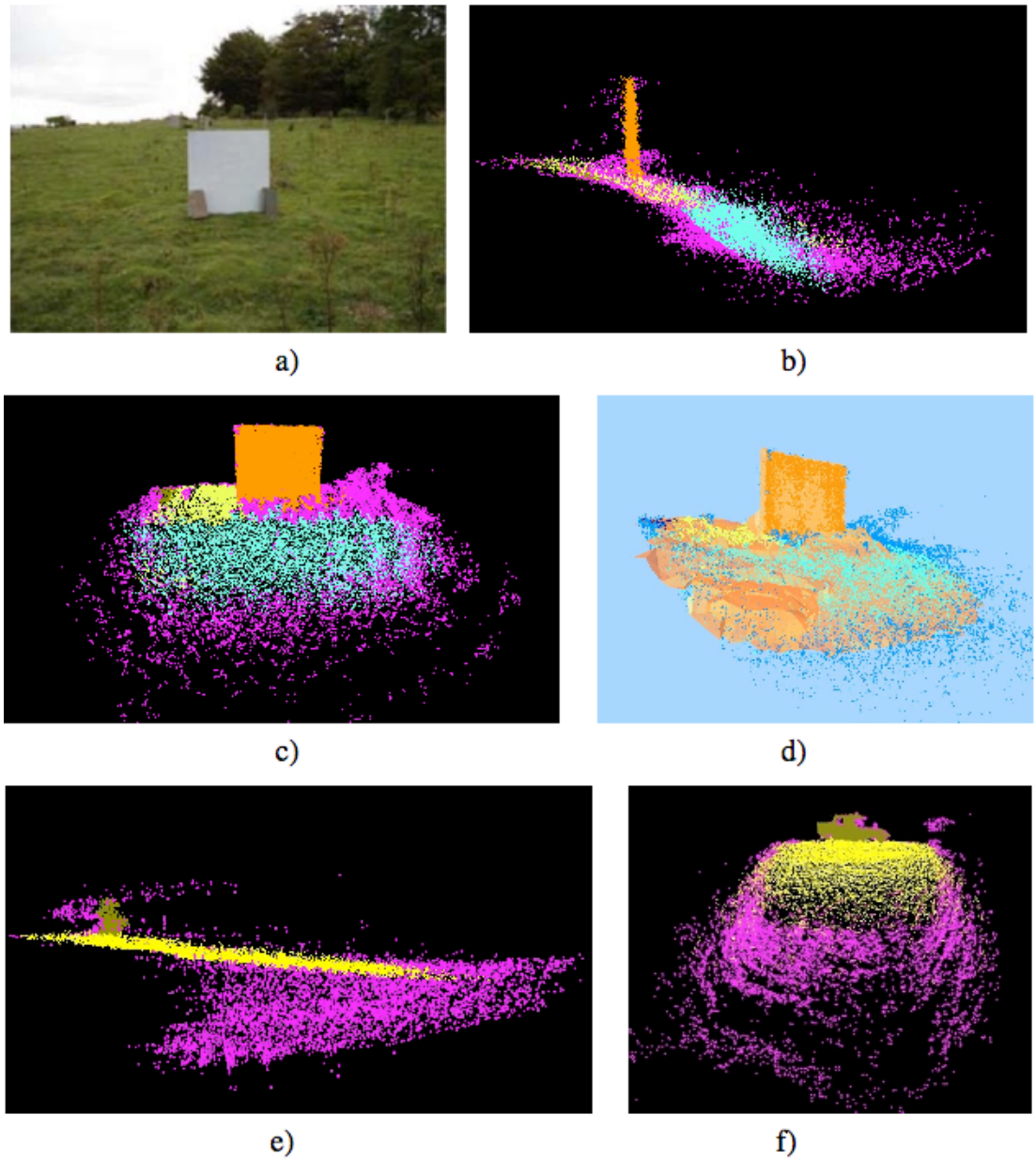
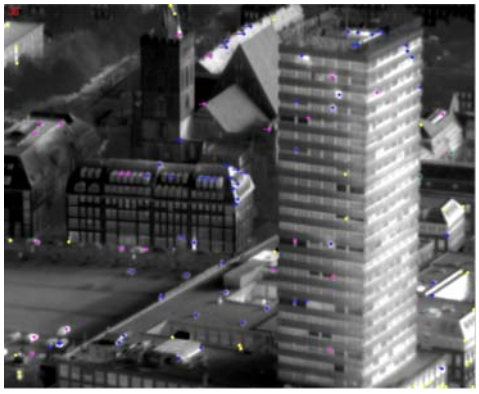


Figure 4-24: Iterated Mahalanobis Segmentation Applied to Burst Illumination LIDAR Data Provided by DSTL-UK NATO SET-118 Collaboration – a) Photo of test scene with target sign in foreground and vehicle in background; b) Side view of segmented sub-clouds around target sign with 'residual label' magenta; c) Front view of segmented sub-clouds around target sign; d) Reconstructed surfaces around target along with point cloud; e) Side view of background vehicle; f) Front view of background vehicle.

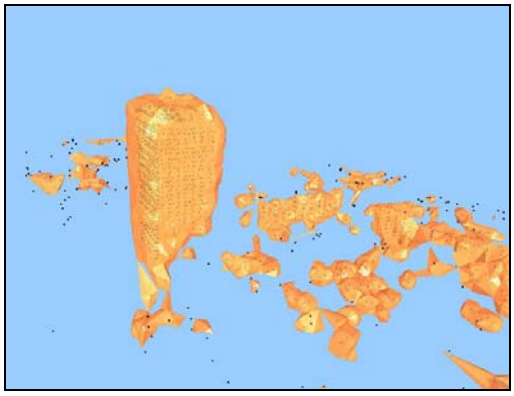
4.4.5 Processing of FOM – Urban SLAM Data Set

German (FGAN-FOM) members of SET-118 provided a 3D point cloud generated from vision-based video using SLAM (Simultaneous localization and mapping). The video was taken from a helicopter hovering around a building. Individual frames were processed for feature points using a corner detector. These points were correlated in adjacent frames using structure from motion techniques from computer vision in order to estimate their 3D positioning, as well as the camera state. Figure 4-25a) is a photo of the imaged building and its surroundings. This was a single frame extracted from the video with superimposed

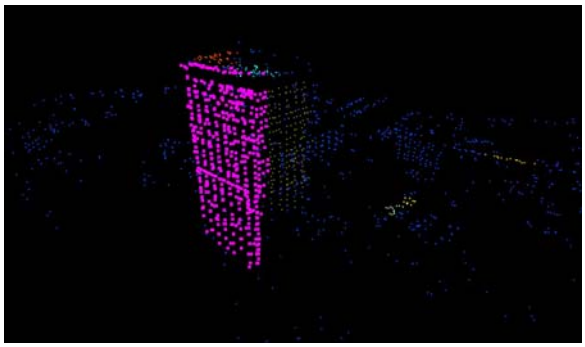
feature points from that frame. Figure 4-25b) shows the full SLAM generated point cloud along with a surface reconstruction using an implicit fit of the heterogeneous sparse point collection. Figure 4-25c) is a rendering of the point cloud after segmentation, where distinct segmented components are labelled with colours (for example, magenta for the front face, orange, yellow, light blue for other components). Figure 4-25d) shows the result of fitting a surface to each component individually and then performing the corresponding Boolean operation to the collection of segments. This provides an acceleration of basic operations such as line of sight and viewsheds of the geometry.



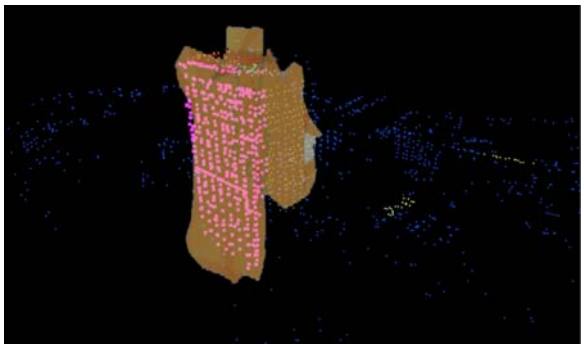
a)



b)



c)



d)

Figure 4-25: Segmentation and Surface Reconstruction of FOM SLAM Data – a) Photo of building with overlaid feature points; b) Point cloud generated from SLAM applied to sequence of feature points from aerial video with approximating multi-resolution surface fit using unsigned distance; c) Iterated Mahalanobis processing of SLAM generated point cloud, where different segments are labelled by colour; and d) Reconstruction of surface by fitting several segments individually and unifying the surfaces.

4.4.6 Defence Research and Development Canada, Valcartier LIDAR Data Collection

SET-118 members from DRDC-Valcartier provided high quality registered LIDAR data acquired by a ground level vehicle. The data set features occlusions, missing data, multiple returns and moderately complex topology. The Mahalanobis procedure was applied to the data set and individual features produced as segments were reconstructed. A sample of the point cloud, the Mahalanobis segmentation, and a resulting surface reconstruction of one segment is displayed in Figure 4-26. Image in Figure 4-26a) depicts the original LIDAR point cloud. Figure 4-26b) is a rendering of the iterated Mahalanobis-segmented sub-clouds labelled by various colours, with Person 2 coloured as magenta. A multi-resolution surface reconstruction is shown for these two sub-clouds in Figure 4-26c)-d), respectively.

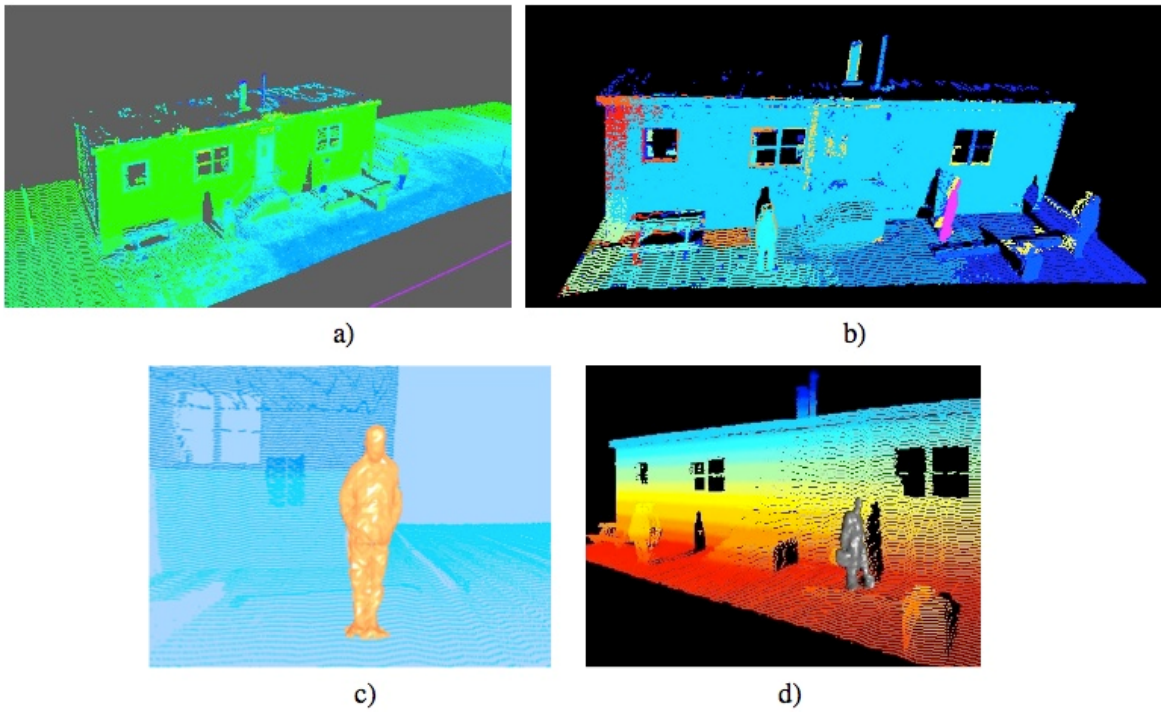


Figure 4-26: Iterated Mahalanobis Segmentation of Scanning LIDAR Data Provided by DRDC Valcartier NATO SET-118 Collaboration – a) LIDAR Point Cloud; b) Segmented sub-clouds labelled by colour; c) Surface reconstruction for person 1; and d) Surface reconstruction for person 2.

4.5 MODEL INSTANTIATION

4.5.1 Semi-Automatic 3D Extraction of Roads

Most of the precision targeting facilities need accurate geospatial information, including vector 3D models of urban infrastructures, which should be extracted from sensing datasets (e.g., satellite imagery) with well-defined accuracy requirements. In a typical 3D reconstruction of an operating scenario, linear features as roads or rivers are the ones most suitable for semi-automatic extraction.

There are different approaches to the problem of "road extraction" from satellite images, depending on the requirements and constraints based upon the end applications. Typically, two approaches are used:

- Automatic extraction where the user usually selects an area of interest and the system extracts all the visible roads in that area; and
- Semi-automatic approach where the user selects the road to extract, by selecting few points.

For our purposes we have adopted a semi-automatic approach, giving the user full control on the road to extract. In particular, the user has to select the start and the end points of the road he wants to extract, and then the software should generate from a stereo-pair of images a list of three-dimensional georeferenced triangles, which precisely model the surface of the road. Furthermore, we require the following constraints:

- To use only RGB data from the image stereo-pair, without any other data (LIDAR information, multi-spectral images, etc.).
- To use a Commercial Off-The-Shelf (COTS) photogrammetric tool, on a Windows-based standard personal computer, for basic stereo-pair handling.

- The final output shall be a vector model composed of a triangle strip, which approximates the road surface with a specific maximum error.

The extraction process is based on the following steps:

- Acquisition of the inputs: the first and last point of the road, the widths in these points (boundaries).
- Detection of the "centreline" of the road.
- Detection of the two sides of the road, using the "Ribbon Snake" or the "Ziplock Snake" algorithm.
- Altitude determination, using cross-correlation of patterns in the stereo-pair.

In the following paragraphs, each step is further detailed.

4.5.1.1 Road Boundary Detection

The first phase of our extraction algorithm requires the user to choose the first and last point of the road he wants to extract. Since the algorithm also needs the road width at the two extremities (the boundary widths), we help the user during this estimation process: as the user moves the mouse cursor along the road, we calculate automatically and visualize in real time the width of the road (see Figure 4-27).

Figure 4-27: Boundary Detection – the Roads to Extract (left) and the Detected Boundaries (right).

At the beginning of the extraction process, the mouse cursor changes in two concentric circles (as shown in Figure 4-27: the bigger one indicates the maximum search area for width detection, the smaller one indicates the minimum threshold to avoid disturbing factors (i.e., cars or middle lines). The radii of the two circles can be freely modified by the user.

The road width is detected by means of the *Canny edge filter* [61] and the *Hough transform line detector* [62]: the Canny edge filter gives us the capability to extract the edges of the road and the Hough transform line detector transforms the edges in a set of lines. Finally, we developed an algorithm (based on some built-up rules) that finds the best couple of lines and calculates the width and the orientation of the road.

4.5.1.2 Centreline Detection

The next phase of the road extraction process aims to determine the road *centreline*. This task is performed by means of the *Steger curvilinear feature detection* [63]. We put the input image into a filter chain, and we

pass the output of the chain to the Steger algorithm (see Figure 4-28). We use a locally-adapted contrast enhancement *Wallis filter* [64] in the chain, to enhance the contrast between road and surroundings.

Figure 4-28: Output of the "Steger Line Detector" Algorithm.

The segments found by the curvilinear feature detection process are elaborated by a *best path finding* algorithm to determine the correct road centreline, using a *gap filling* feature to minimize the effect of partial or total occlusions due to trees, cars, etc. (see Figure 4-29).

Figure 4-29: Road Centreline Calculated by the Path Finding Algorithm.

4.5.1.3 Road Modelling by the "Snake" Algorithm

We model the road using "ribbons" whose sides correspond to the road boundaries. We have chosen the *Ribbon Snake* algorithm to perform this task. A Ribbon Snake [65] is a time-dependent curve defined parametrically as:

$$\vec{v}(s,t) = (x(s,t), y(s,t), w(s,t)), \quad 0 \le s \le 1.$$

Such representation implies that each $v(s_0, t_0)$ is characterized by its width $2w(s_0, t_0)$ and the location of its centre $(x(s_0, t_0), y(s_0, t_0))$. All the centre points compose the centreline of the ribbon.

The snake deforms itself as the time progresses, to minimise an energy functional composed of three terms:

- A geometrical component which controls the snake tension.
- A geometrical component which controls the snake rigidity.
- The image contribution (the magnitude of image gradient). This is responsible for the snake attachment to the road contour.

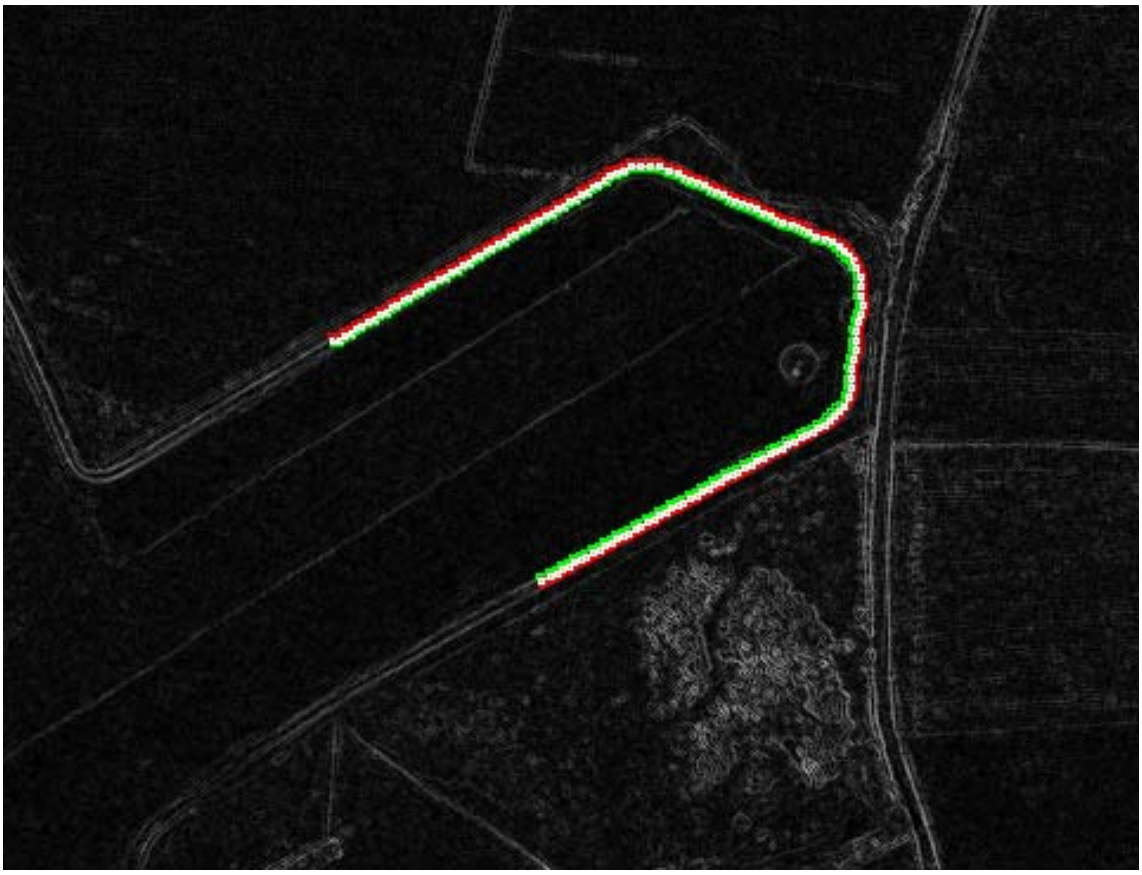


Figure 4-30: Snake Initialization by Centreline and Boundary Widths.

We use the centreline detected in the previous step and the two boundary widths given by the user, as the initial state for the Ribbon Snake. We use the Dynamic Programming implementation for the optimization procedure: it is an iterative algorithm that stops when the total energy of the snake is minimized (considering the energies from all the points). The resulting snake is passed to a *decimation* algorithm that deletes some points, depending on their collinearity properties.

4.5.1.4 Road Modelling by the "Ziplock Snake" Algorithm

In certain cases, the road centreline could not be found, due to strong total occlusions, like building shadows or forests. So, we developed an alternative algorithm to use only in these cases. It does not use the centreline data, but a modification of the original Ribbon Snake algorithm, named "Ziplock Snake" [66]. In this version, the snake is optimized taking into account first the image information near its extremities and then, progressively, toward its centre.

As shown in the image (see Figure 4-31), the snake is composed by two parts, the active part, in which each point is influenced by all the energy contributions, and a passive part, in which the image contribution is disabled. The force boundaries are moved individually, one vertex at the time. A force boundary is advanced when we can verify that the motion of the corresponding active section has stabilized.

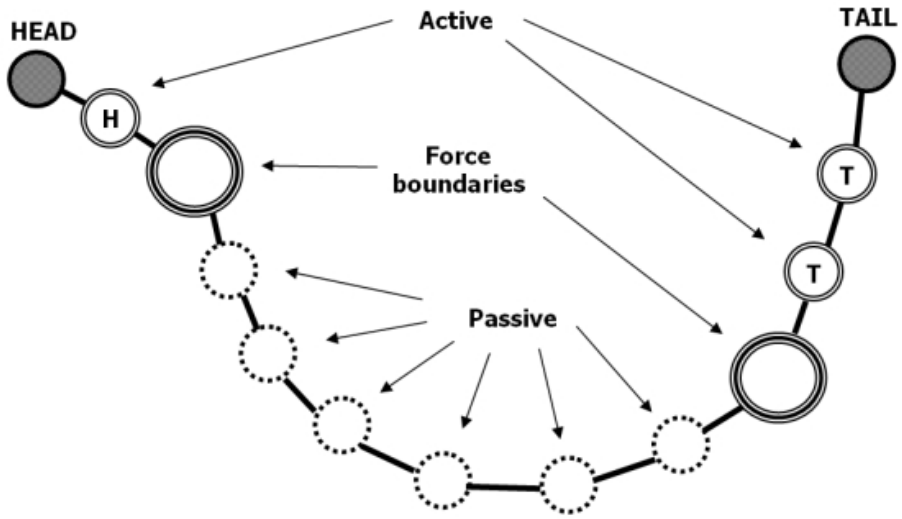


Figure 4-31: Ziplock Snake Elements.

We can see the ziplock approach in this sequence of images (see Figure 4-32): the snake is attached to the image contour in a way similar to a ziplock being closed. Due to the computational expense of the computation required, one should use ziplock snakes to model portions of roads shorter than the ones modeled by the centreline method.

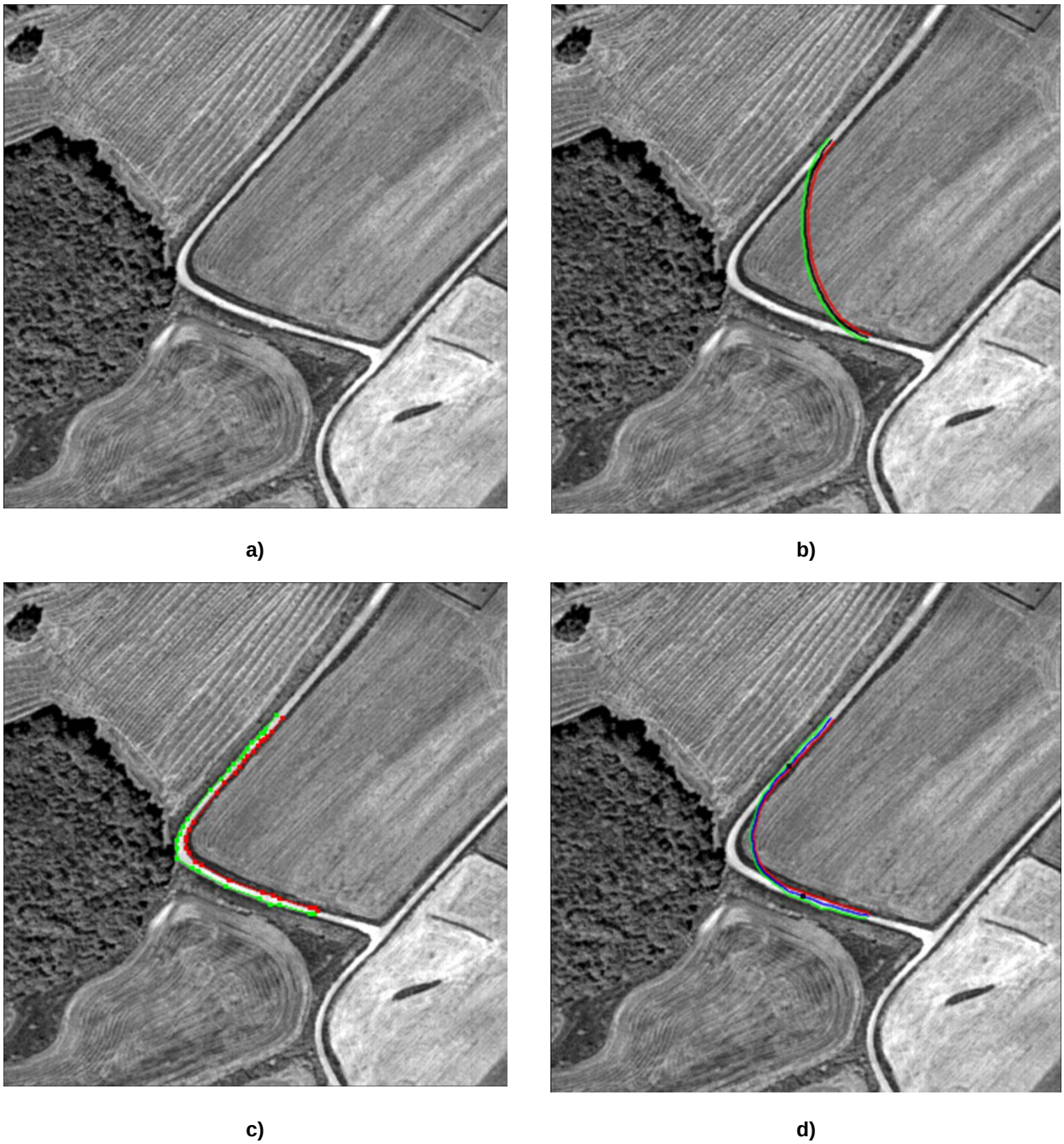


Figure 4-32: Ziplock Snake – a) Input image; b) Snake initialization; c) Snake iteration; and d) Final result.

4.5.1.5 Altitude Determination from Stereo-Couples

- The points we have extracted from the images during the previous steps are georeferenced in latitude and longitude, but their altitude is approximated. In order to assign the right altitude value to each point, we perform the calculation applying a cross-correlation algorithm to the stereo-couple: as the algorithm recognizes the same pattern in the two images, it is possible to determine the correct elevation of the point by triangulation. These altitude values have to be checked, since they could be affected, in some cases, by cross-correlation errors (i.e., due to shadows and/or discrepancies between

the times in which the images were taken). The coherence of these altitude values is automatically verified by the software against a set of criteria and, if necessary, the points in the neighbourhood are analyzed until an acceptable result is reached.

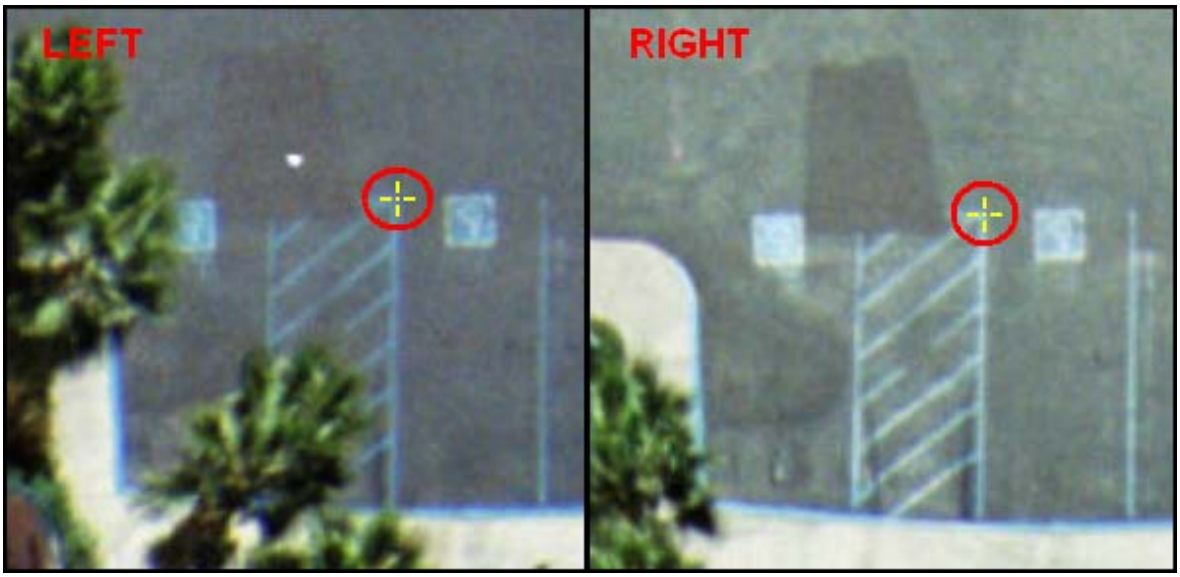


Figure 4-33: Cross-Correlation Example in a Stereo-Pair.

4.5.1.6 Results and Validation

The polygonal models for three test cases are shown in Figure 4-34, Figure 4-35 and Figure 4-36. For each test case, the reference model and the automatically generated road are shown. The reference models were manually extracted and checked for the best accuracy, using three Ikonos satellite stereo-pairs (GSD about 1 meter). An automatic test was performed on the three couples of road models, in order to calculate the horizontal and vertical rms (Root Mean Square) differences between the reference model and the automatic extracted road. Such differences were calculated on both the two sides of each road, using a sampling step equal to 0.01 meters: this means that we have calculated 100 samples for each meter of the reference road. The optimal sampling rate was determined by the stability analysis depicted in Figure 4-37: the rms values reach the convergence when the number of samples per meter is greater than 80.

Figure 4-34: Reference Model #1 (up) and Automatically Extracted Road #1 (above).

Figure 4-35: Reference Model #2 (left) and Automatically Extracted Road #2 (right).

Figure 4-36: Reference Model #3 (left) and Automatically Extracted Road #3 (right).

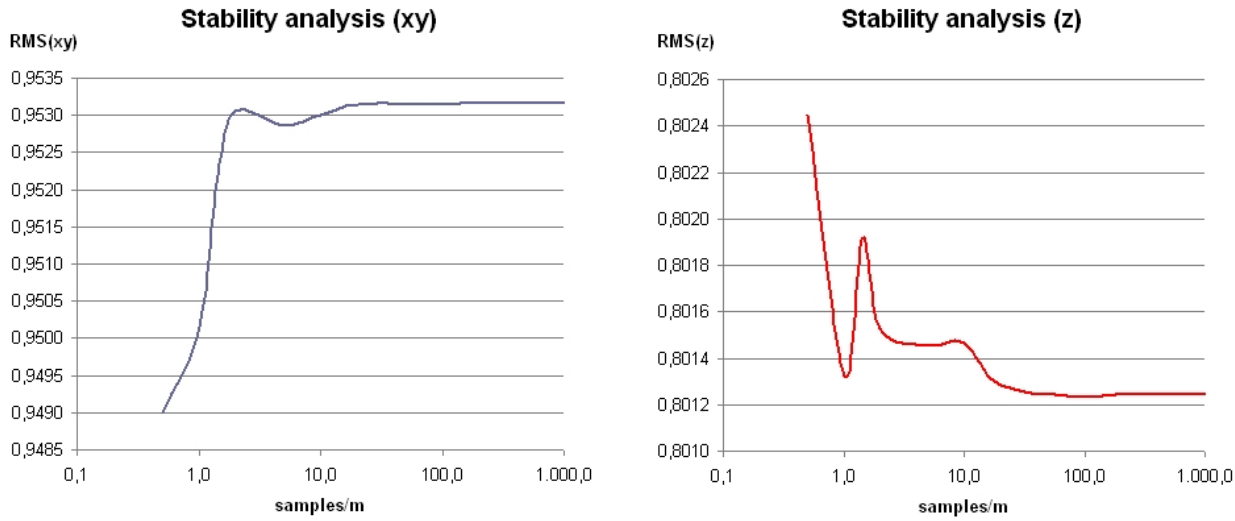


Figure 4-37: Stability Analysis for the xy-Plane and the z-Axis.

In the following table, the horizontal and vertical rms differences are shown, for three test cases (see Table 4-3).

Table 4-3: Horizontal and Vertical rms Differences for Three Test Cases.

Test Case	Horizontal rms (xy-Plane)	Vertical rms (z-Axis)
Road Model #1	0.953 m	0.801 m
Road Model #2	0.619 m	0.218 m
Road Model #3	0.582 m	0.439 m

4.5.2 Building Reconstruction from Multi-Aspect High-Resolution InSAR Data

In Sections 4.1 – 4.3 of this chapter, reconstructions based upon LIDAR sensor data were described. In this section we will give a brief overview of existing reconstruction approaches considering multi-aspect SAR or InSAR data. Subsequently, the magnitude and interferometric phase signature of flat- and gable-roofed buildings under orthogonal viewing directions is analyzed. In the last section our reconstruction approach is described and results based on high-resolution InSAR data are presented.

4.5.2.1 State-of-the-Art

A variety of building reconstruction methods based upon various sensor data have lately been presented in literature. In the following, only the group of approaches based on multi-aspect data is considered and shortly discussed.

4.5.2.1.1 Building Reconstruction through Shadow Analysis from Multi-Aspect SAR Data

The extraction of building dimension by analyzing the building shadow based on a single SAR magnitude image was presented in Bennett and Blacknell [67]. Extending this approach to multi-aspect SAR magnitude, images some ambiguities in the shadow interpretation can be solved and more robust reconstruction results are achievable [68],[69],[70]. The drawback of these methods is given by the shadow analysis itself, which assumes suburban area, no interferences with other objects and flat terrain.

4.5.2.1.2 Building Reconstruction from Multi-Aspect Polarimetric SAR Data

Multi-aspect polarimetric SAR data are considered for building reconstruction by Xu and Jin [71]. First, parallel line segments are extracted from the multi-aspect data by means of a local Hough transform. Second, parallelograms are generated, which contain the layover area of the buildings. Subsequently, a classification takes place to discriminate between parallelograms of direct reflection from the ones of double-bounce signal propagation. In the last step, a maximum likelihood method is used to fuse the multi-aspect information and to reconstruct buildings three-dimensionally. The main demand of this approach is that buildings are isolated, which is difficult in dense urban areas and for higher buildings.

4.5.2.1.3 Building Reconstruction from Multi-Aspect InSAR Data

In Bolter and Leberl [72] the detection and reconstruction of buildings is based on multi-aspect InSAR data by considering InSAR height and coherence. First, the maximum height value of all aspects is chosen and the resulting height map is smoothed with a median filter. Afterwards, potential building regions are generated and minimum bounding rectangles are fitted to these regions. The differentiation between buildings and other elevated vegetation is based on coherence and height map. In a last step, simple building models with either a flat roof or a symmetric gabled roof are fit to the segmented building regions. This approach was further extended in [73] including information from corresponding SAR magnitude data, especially exploitation of building shadows. For these approaches, problems arise if buildings are not isolated or if they are higher than the ambiguity height of the InSAR acquisition.

4.5.2.1.4 Iterative Building Reconstruction Using Multi-Aspect InSAR Data

An iterative building reconstruction approach from multi-aspect InSAR data was presented by Soergel [74]. Based on speckle reduced magnitude images, primitive objects such as edge and line structures are extracted from the slant range data. Subsequently, building hypotheses are set up by generating parallelograms from primitive objects for every aspect. Thereafter, the hypotheses are projected from slant range to ground range geometry in order to fuse the multi-aspect hypotheses. First, the buildings are reconstructed as elevated objects with flat, gabled, or pent roofs as well as right-angled footprints. In order to overcome occlusion effects, building candidates from multiple aspects of the same scene are fused. The resulting building

hypotheses are used as input for a simulation to detect layover and shadow regions. The comparison between this simulated SAR data and the original SAR data shows differences in case of false detections, which are subsequently eliminated by generation of new building hypothesis. The entire procedure is repeated iteratively and is expected to converge towards a description of the real 3D scene. The main drawback of this work is a minimum size of reconstructible buildings. Due to the fact, that the first building hypotheses are assembled in the slant range geometry, very small buildings (e.g., 10 m x 10 m x 5 m – width x length x height) are undetected by this approach. Hence, in our approach [75] the assembly of building hypotheses is realized in the ground range geometry. Furthermore, along with the magnitude signature, an analysis of the interferometric phase signature is considered during the reconstruction approach, which will be described in much more detail in the following sections.

4.5.2.2 Signature of Buildings in High Resolution InSAR Data

In this sub-section, the InSAR signature of flat- and gable-roofed buildings under orthogonal illumination directions is shown and analysed. Especially the signature parts relevant for our building reconstruction are characterized in the magnitude and interferometric phase signatures.

4.5.2.2.1 Magnitude Signature of Buildings

The appearance of buildings is characterized by the side-looking viewing geometry of the SAR sensors and range measurements. In the first row of Figure 4-38 the optical signature and, in the second, the typical SAR signature parts – layover, corner, roof and shadow are schematically shown, whereby the grey scale values represent the expected magnitude value in the SAR data. The layover results from the integration of a different backscatter signals (e.g., ground, building wall and roof) into the same range cell. The corner is caused by double bounce reflection between ground and building wall. The roof area is characterized by a single backscattering of the building roof. The ground behind the building is partly occluded and appears then as dark shadow region.

In the third row of Figure 4-38, the real SAR magnitude signature of the two buildings is given by considering the orthogonal illumination directions. The signature of the flat-roofed building changes radically between both data sets. The single backscattering of roof area is not observable in the second data set. Similar effects are observable for the gable-roofed building. The layover area is sub-divided in two parts, which are characterized by different backscatter contributors. The first part, the bright line, results from sloped roof, wall, and ground, and the second darker part from wall and ground only.

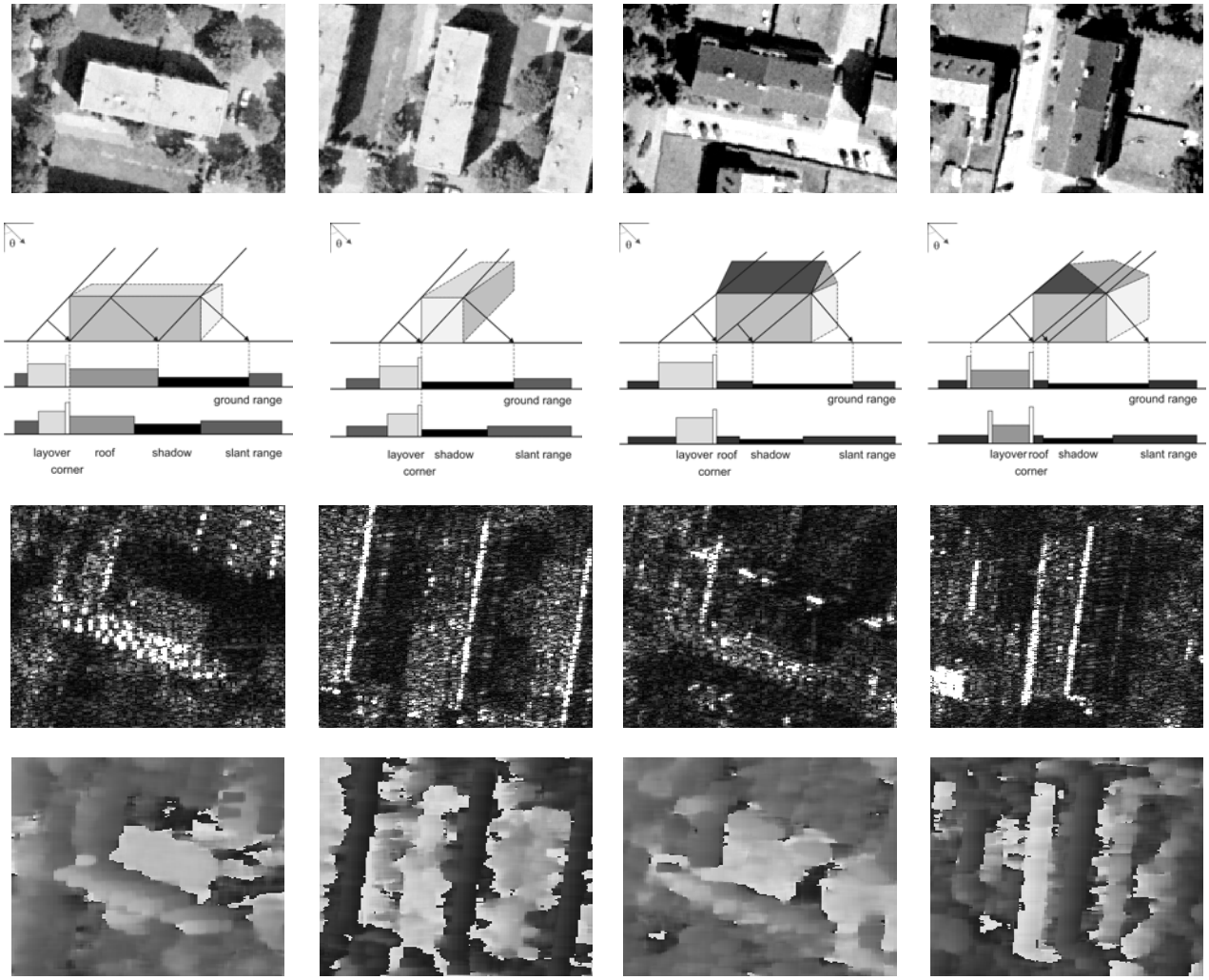


Figure 4-38: Appearance of Flat- and Gable-Roofed Buildings Under Orthogonal Illumination Conditions – Optical image (first row); Schematic view (second row); SAR magnitude data (third row); and InSAR phase data (fourth row).

Next to the illumination direction, also the building dimensions and the off-nadir angle θ influence the building signature, which was discussed for flat-roofed buildings in [75] and for gable-roofed buildings in [77]. For our purpose of small building detection, the signature analysis pointed out that the corner lines and double lines in the magnitude data are the most stable and dominant features. Hence, our building recognition and reconstruction is based primarily on bright lines segmentation.

4.5.2.2.2 Interferometric Phase Signature of Buildings

Beside the magnitude pattern, the interferometric phase signature of buildings is characterized by layover, multi-bounce reflection, direct reflection from the roof and shadow, too. In Figure 4-38 (fourth row) the variation of the InSAR phase signature due to different building types and illumination properties is shown. In general, the phase values of a single range cell result from a mixture of the signal of different contributors. Hence the final InSAR height of an image pixel is a function of the heights from all objects contributing signal (e.g., heights from terrain, building wall, and roof) to the particular range cell. The layover region in the InSAR phase data is also called front-porch region and shows a downward slope from close range to far range in slant range direction. For the flat-roofed building signature, this is caused

by the two constant (ground and roof) and one varying (wall) height contributor. The significant corner position in the magnitude profile shows in the phase profile a phase value nearly similar to local terrain phases. This is caused by the sum of the double-bounce reflections between ground and wall, which have the same signal run time as a direct reflection at building corner point. The subsequent single response of the building roof leads to a constant trend in the phase profile. From the shadow region no signal is received so that the phase is only characterized by noise. Focusing on the illumination direction, similar effects are observable between the magnitude and phase data. The orthogonal illumination configuration given in the second column leads to a dominant layover area, but no constant phase area from the building roof does exist due to the small building width.

The phase signature of the gable-roofed building under the given illumination direction (third column) is mainly dominated by the backscattering of ground and building wall. Reason for less developed response of the building roof is the roof orientation away from the sensor. In the orthogonal case (fourth column) the response of the roof dominates the layover signature. As a consequence, the layover maximum is much higher than for the other illumination direction. The shape of the layover phase profile is determined by the off-nadir angle, the eave, and the ridge height. For example, a strong steep slope leads to a high gradient in the phase profile. A more detailed analysis of flat- and gable-roofed building phase signatures is given in [76] and [77].

Summing up, for our purpose of small building detection, the signature analysis of InSAR data pointed out, that the corner lines and double lines in the magnitude data are the most stable and dominant feature for building detection. Hence, our building reconstruction is based primarily on bright lines segmentation. Furthermore, the layover region in the phase data contains a lot of geometric building information, which can be very helpful during the 3D reconstruction of gable-roofed buildings and the post-processing of flat-roofed buildings.

4.5.2.3 Building Reconstruction Approach

In the following, our approach of building reconstruction based on multi-aspect high-resolution InSAR data is described. The full workflow is given in Figure 4-39 and main processing steps are described briefly.

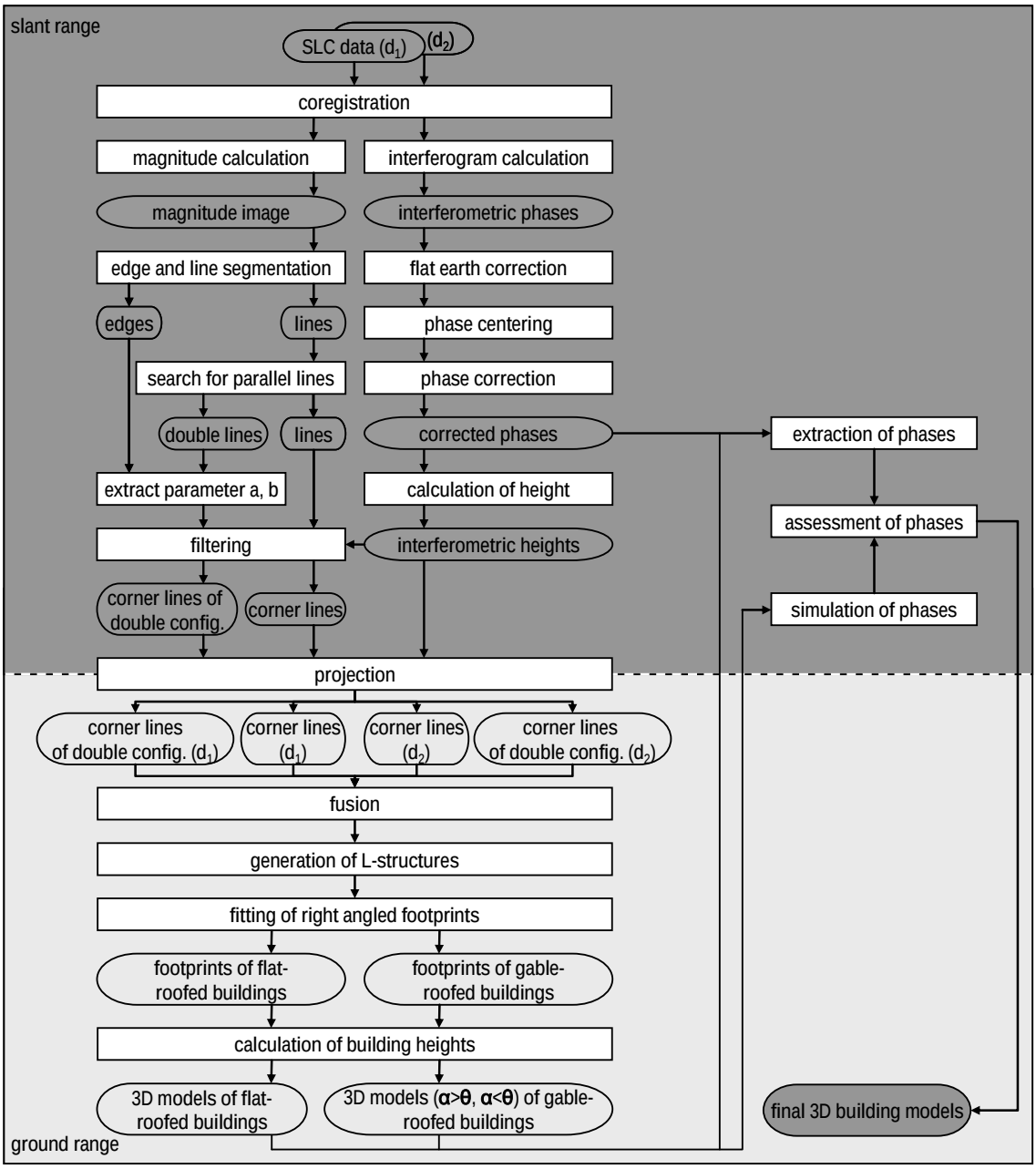


Figure 4-39: Workflow of Algorithm.

We assume as prior knowledge of the buildings that they have rectangular footprints, a minimum building coverage of 8 m x 8 m x 4 m (width x length x height), vertical walls and a flat or gable roof. The recorded InSAR data have to contain acquisitions from at least two aspects spanning an angle of 90 degree in the optimal case in order to benefit from complementary object information.

4.5.2.3.1 Pre-Processing

The pre-processing of the data starts in the slant range geometry and contains the sub-pixel registration of the interferometric image pairs and the calculation of the interferograms. This interferogram generation includes multi-look filtering, followed by flat earth compensation, phase centring, phase correction, and height calculation.

4.5.2.3.2 Extraction of Building Features

The extraction of building features contains:

- Segmentation of primitives.
- Extraction of gable-roofed building parameters.
- Filtering of primitives.
- Projection and fusion of primitives.

Based on the magnitude images, edge and line primitives are segmented by using an adapted detection operator to [78] and [79]. Subsequently, based on the segmented bright lines "double line" pairs are assembled and two geometric building parameters of gable-roofed buildings are calculated. More details are presented in [80]. Thereafter, the primitives are filtered by exploiting the characteristic interferometric heights at the corner line position. The final corner lines have to show a mean InSAR height value similar to mean terrain height. In the next step, the corner lines of each aspect are projected from the individual slant range to the common ground range geometry. The fusion of all multi-aspect primitives provides a common group of building primitives.

4.5.2.3.3 Generation of Building Hypotheses

The generation of building hypotheses contains the generation of building footprints and the calculation of building heights. The building footprints are generated from the common group of building primitives by assembling pairs of lines, which form an L-structure. Based upon this right angled footprints and the footprint hypotheses are passed. The subsequent height estimation considers the building roof type, whereby the results of "double line" segmentation are used to distinguish between flat- and gable-roofed building hypotheses. More detailed information as well as the equations for the height estimation are given in [80].

4.5.2.3.4 Post-Processing of Building Hypotheses

The final post-processing of the building hypotheses is realized by a detailed analysis of the building InSAR phases. This analysis is based on the comparison of real InSAR phases and simulated phases [76] received from the generated hypotheses. First, ambiguity problems in the reconstruction of gable-roofed buildings can be solved in this way. Second, the resulting oversized footprints caused by adjacent trees or fences can be corrected by the comparison step. A detailed description of both post-processing steps in given in [80]. After the assessment step the final 3D building results are created.

4.5.2.3.5 Results of Building Reconstruction

The results of the building reconstruction are presented in Figure 4-40. As test area the city of Dorsten (Germany), characterized mainly by residential flat- and gable-roofed buildings, was chosen. The InSAR data were acquired by the Intermap Technologies X-Band sensor AeS-1 [81]. The data show a spatial resolution of about 38 cm in range and 16 cm in azimuth direction and were taken by an orthogonal viewing configuration.

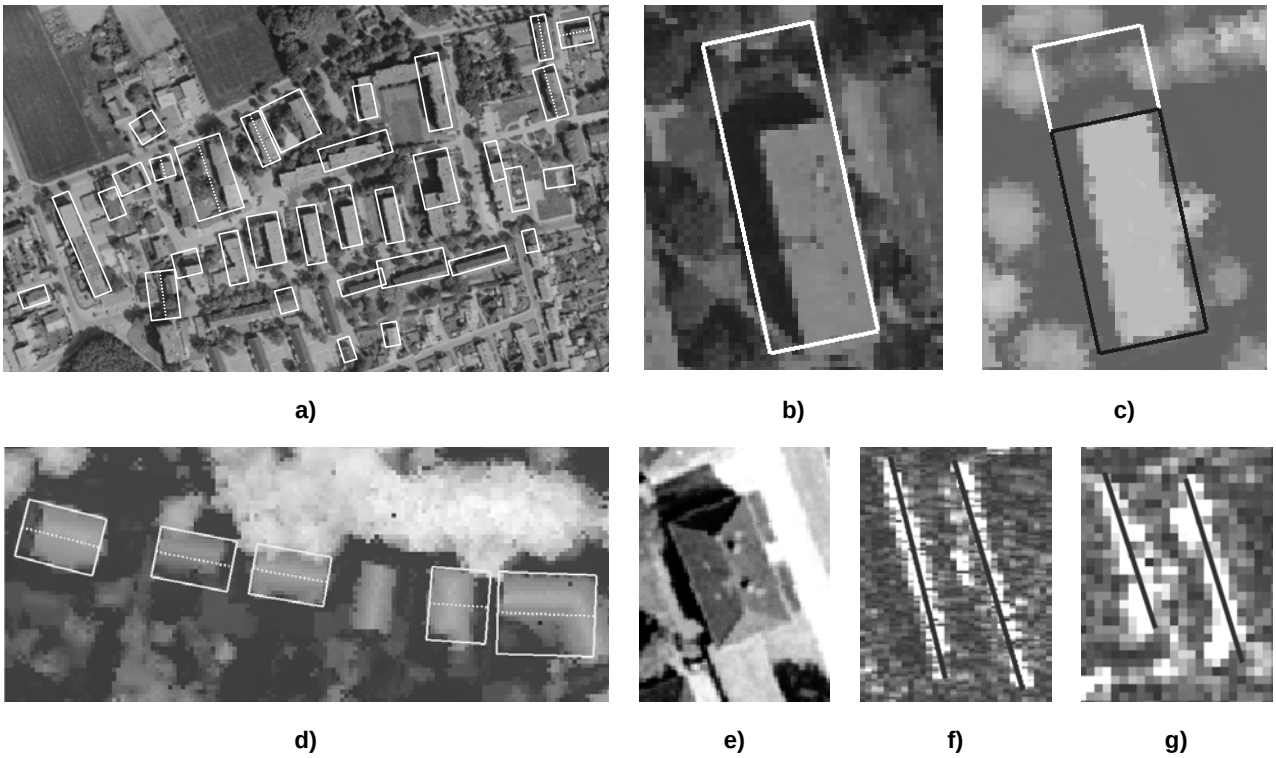


Figure 4-40: a) Results of Mainly Flat-Roofed Building Detection; b) Results of Gable-Roofed Building Detection; c) Oversized Building Hypothesis; d) Corrected Building Hypothesis; e) Optical Data of Hip-Roofed Building; f) Detection Result of AeS-1 Data; and g) TerraSAR-X Data.

In Figure 4-40 the results of the flat- and gable-roofed building reconstruction are shown. The majority of the buildings given in (a) are well detected and shaped. The less detection in the lower part of the scene is caused by missing overlap between the multi-aspect data. Furthermore, our building recognition failed if trees or buildings were located too closely to the building of interest. Some of the reconstructed footprints, especially the row of buildings in the middle of the scene, are larger than ground truth, due to too long segmented corner lines caused by signal contributions of adjacent trees or fences.

The second part (b) shows good results for the gable-roofed building detection, too. The ridge orientation of four buildings is well detected. The fifth building shows a hip roof instead of the assumed gable roof. The final 3D results of this building group are given in [77].

Focusing on the presented post-processing results, the oversized building footprint (c) is well corrected and given in (d). In detail, the building length is well corrected from 50.7 m down to 36.9 m (ground truth 36 m). Also the building height (from 9.8 m to 11.4 m – ground truth 13 m) and the height standard deviation (from 4.0 m to 3.3 m) inside the building footprint are strongly improved.

Summarizing the results, there is still room for improvement. On one hand, the processing improvement is possible to achieve better results after the post-processing step or to obtain a higher completeness of building recognition by combining more than only two aspects. On the other hand, the adaptation of this approach for high-resolution airborne data (e) to the new generation of high-resolution space borne data (f) is desirable, especially by regarding these first results of “double line” detection.

4.5.3 Extraction of 3D Information of Individual Objects with Monoscopic Techniques

Monoscopic techniques are a generic methodology best fitted to extract height information on isolated features such as buildings. The monoscopic technique is based on the photo-interpretation of a single

image instead of a pair of images utilised in the stereoscopic approach. For the specific purpose of 3D feature extraction, the third dimension can be derived from auxiliary information that is present in the image. The methods proposed to obtain the height of the 3D features provide an estimation of the height, whose precision depends on the conditions of the image, as it will be detailed in the following paragraphs.

Two methods have been analysed for the estimation of the features in the monoscopic photo-interpretation:

- The shadow of the buildings and the sun elevation. This information allows deriving the height of the buildings.
- Photo-interpretation of the type of building. In this case the height of the building cannot be measured directly in the image, but it is determined by estimation.

These methods utilise very simple algorithms but can be very useful for security applications, where it is more important to provide methods that allow a rapid 3D extraction than obtain extremely accurate results.

4.5.3.1 Estimation of Height of a Feature from the Shadow of the Feature

This method consists in measuring the shadow of a feature (normally a building or any other feature which is high enough to apply this method) in the image. The height of the feature is calculated as a function of the shadow length and sun elevation angle.

The sun elevation angle is usually included in the metadata accompanying the images. The calculation of the height of the feature is performed by solving the trigonometric problem shown in the next figure. This simple method allows extracting features that are high enough to project a shadow that can be measured with precision. The following figure also shows a QuickBird satellite image example from an airport control tower in Madrid.

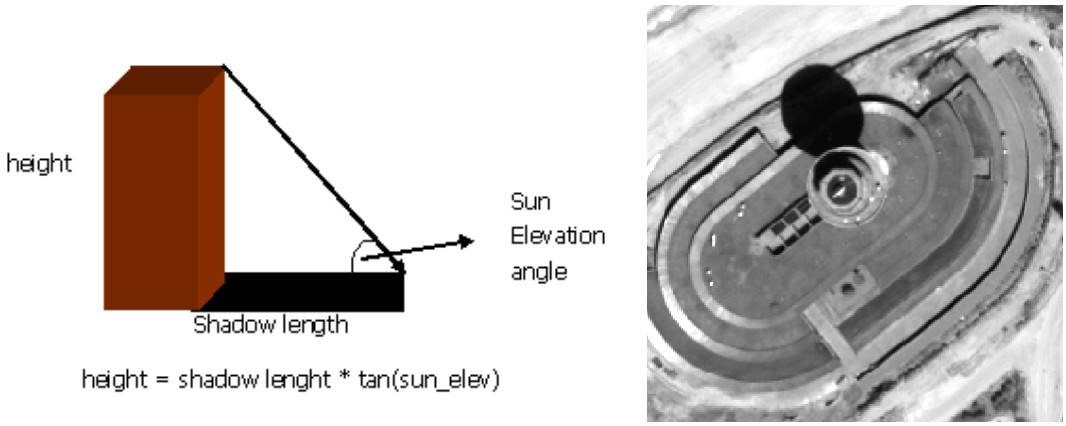


Figure 4-41: (Left) Height Calculation from the Shadow; and (Right) Exemplary Case of Application Under Ideal Conditions, i.e., Measurable Shadow, of an Airport Control Tower.

However, there are typically two situations which jeopardize the application of this technology, i.e., two ways in which shadow cannot be measured: when the shadow is in occultation or when we face a distributed or a too complex object.

When the image acquisition azimuth angle is very similar to the sun elevation, the shadow is occulted by the buildings. In this situation, the sun the target scene and the satellite are aligned and the shadow area is eclipsed by the 3D feature under observation. In the next figure we can find an example of this situation, where the shadow is mostly hidden by the buildings and scarcely can be measured.

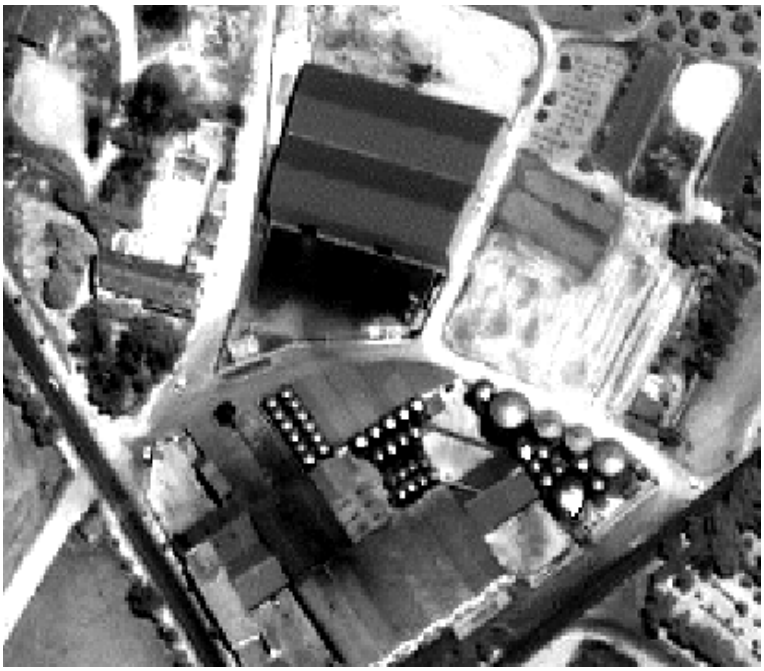


Figure 4-42: Satellite Image with No Visible Shadows.

When the 3D object is very complex and the shadow is not projected over flat terrain, but over another 3D object, the method is not applicable. This is the case when instead of analysing a simple and isolated feature, we analyse a cluster or a distributed object. The shadow is not flat and although it could eventually be measured, the basic geometry model is no longer valid. The following figure shows an example of a compact urban area. Since the houses are aggregated into blocks, information cannot be extracted on single buildings since there are no measurable projected shadows. It is only possible to extract information at a block level. The figure also shows the automatic borders extraction (a typical step used for the extraction of the shadows) where we can see the insurmountable difficulty to interpret individual shadows.

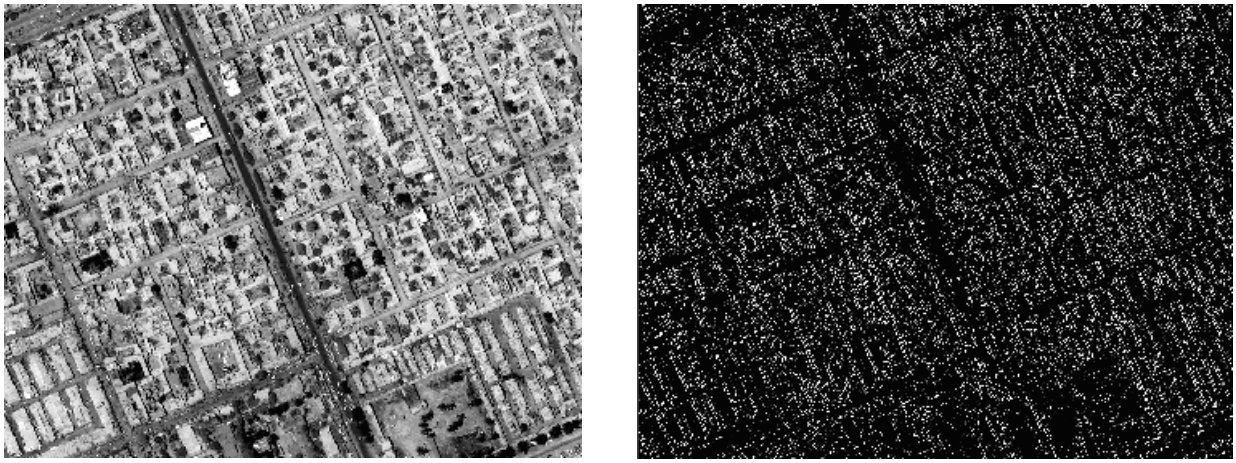


Figure 4-43: Exemplary Situation Where the Shape from Shading Method Cannot be Applied – Left: Satellite image showing complex 3D urban structure where the buildings are aggregated into blocks; Right: Border extraction showing the difficulty to identify shadows at building level. Ikonos image courtesy of SET-118 group.

4.5.3.2 Features Height Estimation by Means of Photo-Interpretation

A different approach when no other method can be applied due to the characteristics of the images (no stereo pair is available and the shadow is not visible) is the estimation of the height of the buildings by photo-interpretation of the shape of the building. The result of this method is only valid for applications not requiring a very accurate measure of the height. This is a fully man assisted method where a photointerpreter needs to identify the outline of the building and estimate the height regarding the shape of the building. For example, the building shown in the following picture can be estimated to have four floors, therefore the height can be estimated to be: 4 + (4 * 3) = 16 meters.

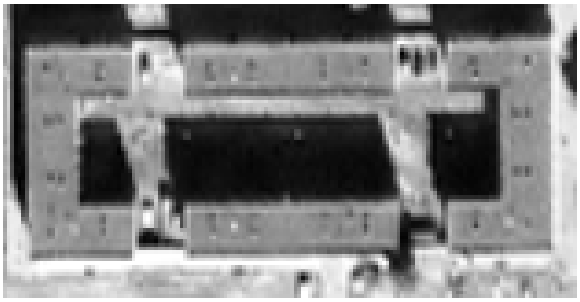


Figure 4-44: Building Example Where the Height is Extracted from Image Knowledge.

The error originated by this method can be very high but, when there is no other available information to estimate the height, it can be enough for certain applications, such as, for instance, the 3D scenarios for simulation applications.

4.5.3.3 Pros and Cons

The advantages of the monoscopic approach are the following:

- It is easily applicable to satellite images where very accurate sun angles measurements are available in the metadata.
- The 3D data extraction is very easy with this method and no specific training is required.
- Specific tools can be developed to improve the extraction of 3D features applying this method.

The disadvantages are:

- The accuracy of the extracted information is not very high; therefore this can limit some applications. This method may be acceptable for the generation of 3D scenarios for simulation where not very high accuracy is required.
- This method cannot be applied to all satellite images: only those images having specific conditions in the sun angles and acquisition angles.

As far as the photo-interpretation method concerned, it requires a previous knowledge of the structures of the buildings that appear in the images in order to be able to estimate the height. The accuracy of the height estimation is low, but it can be enough for applications that require rapid 3D extraction methods and where accuracy is not a relevant requirement to be satisfied.