# FACADE RECONSTRUCTION WITH GENERALIZED 2.5D GRIDS

Jérôme Demantké, Bruno Vallet and Nicolas Paparoditis.

Université Paris Est, Laboratoire MATIS, IGN 73, avenue de Paris 94165 Saint-Mandé, Cedex – firstname.lastname@ign.fr

## **Commission III/2**

KEY WORDS: Geometric, Surface, Modelling, Urban, Building, LIDAR

# **ABSTRACT:**

Reconstructing fine facade geometry from MMS lidar data remains a challenge: In addition to being inherently sparse, the point cloud provided by a single street point of view is necessarily incomplete. We propose a simple framework to estimate the facade surface with a deformable 2.5d grid. Computations are performed in a "sensor-oriented" coordinate system that maximizes consistency with the data. the algorithm allows to retrieve the facade geometry without priori knowledge. It can thus be automatically applied to a large amount of data in spite of the variability of encountered architectural forms. The 2.5d image structure of the output makes it compatible with storage and real-time constraints of immersive navigation.

# **1 INTRODUCTION**

#### 1.1 Motivations

We aim to automatically reconstruct a fine photorealistic facade model (lod3). In this paper, we focus on the geometrical modeling. The geometry is extracted from lidar data and has to be in accordance with the optical images acquired in the same time and used for texture mapping. Apart from being photo-consistent, the model has also to be sufficiently compact for scaling up.

#### 1.2 Related Work

Facade Modeling is a vastly studied topic (Musialski et al., 2012b). Numerous articles offer to reconstruct the facades using knowledge based approaches, looking for semantic objects such as windows (Pu and Vosselman, 2007), (Wang et al., 2011), assuming they are regularly arranged in a grid structure (Müller et al., 2007). The structure information of facades is considered as necessary for a good reconstruction (Becker and Haala, 2009). Indeed, the search for well-defined objects (size, shape ...) is not obvious in point cloud data (PCD). Grammar rules can be used to palliate the lack of information. However the apparent simplicity of the facade structures, and the underlying grammars is actually complex to automatically detect, to the extent that an operator is deemed necessary to get the desired results (Musialski et al., 2012a).

Other methods use more general assumptions like the predominance of right angles. Chauve et al. (2010) proposes to build a 3d model from images by an arrangements of planes, adding horizontal or vertical planes where there is an absence of information. Although such approaches are adapted to human constructions, objects that do not fit the model cause strange artifacts (Furukawa et al., 2009).

Another line of research is the detection of repetitions and symmetries (Pauly et al., 2008). The repetitions that are not necessarily sequenced according to a grid structure (Shen et al., 2011) may form a kind of low-level grammar. Detecting repetitions require to deal with the uneven sampling of the ranks scan (Friedman and Stamos, 2011). Density varies depending on the distance of echoes to the sensor, but also on occluders. Facades are partially



Figure 1: Point density depends on sensor location and orientation (left), and on surfaces geometry (middle). It is also decreased by occluders like trees in front of facades (right). Thus it depends both on acquisition (viewing angle, scanning frequency ...) and scanned objects. Nevertheless, point density is the essence of information given by PCD.

hidden, and density decreases toward the top of the buildings. Instead of replacing missing parts by flat or smooth surfaces, some papers offer to "consolidate" and fill the holes thanks to the repetitive structures (Zheng et al., 2010), (Nan et al., 2010). But here again, the detection of repetitive elements in PCD is complex and requires manual assistance (Li et al., 2011).

Although such knowledge based approaches are promising, Detecting facade structures and grammars remains difficult, as evidenced by the need for human intervention. In addition, such methods cannot handle old style facades, complex ornaments, and any unexpected object which does not match the model. Maybe hybrid reconstructions (surface/shape) are more appropriate to these contexts and could help the facade structure to emerge (Lafarge et al., 2013).

We opted for a low level approach, consistent as possible with the data. Indeed, we believe that a processing step is missing in facade modeling. Trying to detect structure into PCD imply to manage the density variation that can both depend on underlying objects and acquisition context (fig:1). A more "acquisition independent" data is needed. A surface may fulfill this role. Many approaches are looking for the facade surface (Deschaud, 2010), but start with an irregular mesh and fill the holes in a second step. Our approach assumes the presence of a continuous surface that is iteratively adjusted toward the 3d points.

Carving approaches use the information provided by the lidar sensor. Even without knowing the position of the source (Shalom

et al., 2010), it is possible to constrain the surface reconstruction by estimating the angle of view of the sensor. Surfaces thus generated (Hadsell et al., 2009) ensure consistency with the acquired data. Rather than a purely carving method, we propose to adopt the sensor point of view. Namely, we place ourself in a coordinate system adapted to the acquisition method. Using such a coordinate system allows efficient and accurate results, as for the calculation of normal (Badino et al., 2011). The benefits and challenges of using the acquisition coordinate system are well shown by Frueh et al. (2005) in particular, they explain the reversal scan order that causes topological problems when trying to mesh points according to acquisition order. Their choice is to work with depth images from path segments that contain no reversal order. In section 2.1, we will discuss the use of such depth images where each pixel represents a lidar point (fig:3). We also propose to reconstruct a 2.5d grid that leverages the acquisition geometry and overrides the topological problems, but our approach allows more freedom because one pixel can contain zero, one or several points, and the grid can be computed from any path segment.

## 1.3 Overview



Figure 2: Workflow of grid fitting to the primitives.

A 2.5d grid is fitted to the geometrical primitives. In the 2.5d grid coordinate system, any 3d point is defined by u and v that locate the point on the grid, and h that is the depth from the u-v plane. The main steps and the inputs of the algorithm are depicted thereafter and in figure 2.

**Initialization.** A vertical rectangle approximating the main facade wall is calculated as in (Demantké et al., 2012). It provides the u-v plane and the grid boundaries. The 2.5d grid is placed on this rectangle and the depth of each grid pixel is set to zero.

**Primitive accumulation in the grid pixels.** The primitives (lidar points, triangles or line segments) are projected onto the u-v plane in order to accumulate the primitive depths h into the grid pixels: for any kind of primitive, whenever the intersection between a primitive and a pixel is not empty, the primitive depth is accumulated in the pixel.

Iterative deformation of the grid. The grid is then iteratively deformed to draw near the primitives. An optimal h is searched for each pixel according to the depths of the primitives that are projected in, and according the neighboring pixel depths. Only the h coordinates are changed, the pixels therefore move along their iso-uv. This process is performed several times in order to converge to a satisfactory solution, despite bad initializations, outliers and empty pixels. The strategy for pixel depth computation serves to handle cases where many primitives have been accumulated in one pixel. The depth interpolation for empty pixel is made locally, according to a smooth parameter explained in section 3. Each input is independent from the others. As well, the strategy for pixel depth computation can be chosen separately,

however if the grid resolution or some other parameter is changed between two iterations, the primitive accumulation must be performed again.

# 2 COORDINATE SYSTEMS

The choice of the 2.5d grid coordinate system and the corresponding projection is important because it affects the primitives accumulation and it determines the direction of the pixel displacements (orthogonal to the u-v plane if Cartesian coordinate system). Our first intuition was to use a Cartesian coordinate system. This favors perpendicular angles, which are prevalent in human buildings. However, it causes problems that could be avoided using a more "sensor-oriented" coordinate system. That is why we propose a generalization of the orthogonal projection called "prismatic" projection and the associated coordinate system. The prismatic coordinate system can be adapted to the facade and the laser beam orientations.

In this section, we analyze the sensor coordinate system corresponding to an acquisition with a rotating laser, scanning vertically and perpendicular to the trajectory. We list the benefits and the limitations of working in such a coordinate system. Then the prismatic coordinate system is described. Finally, orthogonal and prismatic coordinate systems are compared in the section 2.3.

## 2.1 Sensor coordinate system: relevance and limitations

Finding topological structure of objects from an unorganized point cloud is complex. The most probable surface is often searched, leaving decisions of point linking and hole filling to thresholds or assumptions. In fact, the objects are not sampled from every point of view. The sensor generally sees only one side of the objects, whereas a hidden part remains unknown and impossible to reconstruct. The acquisition result is thus a partial view of the scene. Rather than detecting the whole object volumes, the lidar discovers an empty volume between the sensor and the objects, "carved" by the laser beams. The object surfaces stop the laser beams and bound the sensor visibility area. Lidar echoes are thus located on object surfaces, and more generally at the interface between what is seen and not seen by the sensor. Assuming that this interface is a surface, reconstructible surface pieces (corresponding to the visible part of the objects) are also pieces of this interface. Therefore, we consider this interface and the ways to reconstruct it. This is an image in sensor projection. Points can be simply and rapidly meshed following the neighborhood relationships in sensor coordinate system. This "naïve" meshing has some shortcomings, but more sophisticated techniques adopting the sensor viewpoint may permit to benefit from sensor coordinate system.

Our acquisition vehicle is equipped as in (Paparoditis et al., 2012). The laser scans perpendicularly to the trajectory, with a vertical angle  $\Theta$ . Let P(x, y, z) an echo acquired at  $\tau$  time,

$$P(x, y, z) = S_{\tau} + R\cos\Theta \overrightarrow{U_{\tau}} + R\sin\Theta \overrightarrow{V}$$
(1)

 $S_{\tau}$  is the position of the sensor at  $\tau$ . R is the distance between  $S_{\tau}$  and P.  $\Theta$  is the angle between the horizontal plane and the laser beam. V is a vertical unit vector.  $U_{\tau} = T_{\tau} \times V$ , with  $T_{\tau}$  the unit vector of the vehicle displacement at  $\tau$ . Each term has specific features that imply vari-

Each term has specific features that imply various accuracies and value distributions as shown in table 1.



ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume II-5/W2, 2013 ISPRS Workshop Laser Scanning 2013, 11 – 13 November 2013, Antalya, Turkey

	$S_{\tau}$	R	Θ
Accuracy de-	Positioning	Lidar mea-	Mechanical
pends on	system of the	surement	measurement
	vehicle		
Accuracy is	-	+	+++
Density	Vehicle tra-	Scanned ob-	Lidar firing
depends on	jectory and	jects distance	frequency
	speed	and angle	
Distribution	If constant	If smooth	Yes
regularity ?	speed	surfaces	

Table 1: Accuracy and Distribution of  $S_{\tau}$ , R and  $\Theta$ .



Figure 3: The regular mesh in  $(\tau, \Theta)$  space, folding on itself in the (x, y) space.



Figure 4: Folding of the regular mesh (purple) due to the trajectory curvature.

The lidar sweep produces points at regular intervals of time  $\tau$  and firing angles  $\Theta$ . Hence, point distribution in  $(\tau, \Theta)$  space is very homogeneous and structured. A regular mesh appears if each point is linked with the next point acquired, and with the point on the next scanline that has the same  $\Theta$ . Such a mesh may fold on itself when projected into (x, y, z) space (fig: 3). This is caused by the reversal scan order as shown in fig: 4. The curvature of the vehicle trajectory is the source of laser beam crossings that make points out of sequence along  $\tau$ , whereas the order is always kept along  $\Theta(\text{fig: 5})$ . In addition to this order reversal problem, the neighborhood relationships established according to the sensor geometry does not necessarily make sense in 3d space. Some objects are not surfacic, like trees, generating scattered points. The windows, both transparent and reflective, induce the detection of several surfaces for the same  $(\tau, \Theta)$  area. In these cases, the regular mesh is no longer adequate to represent the data. Mention may also be made of the multi-echoes that share the same  $(\tau, \Theta)$ , and of the laser shots oriented toward the sky that generate "no returning" beams. In sum, the nearest neighbors in sensor geometry are not necessarily the most suitable, because the sensor geometry is trajectory dependant (mesh folding), and because the geometry of the underlying objects may be too complex or sub-sampled. Nevertheless, it seems interesting to work in sensor coordinate system that highlights the almost-regular and almost-2.5d data structure. The main hurdle is the absence of bijection due to the possible laser beam crossings. That is why we propose the prismatic coordinate system that imitates the sensor coordinate system: the iso-uv are aligned along the laser beams as far as possible, while ensuring a bijection between (x, y, z) and (u, v, h): no iso-uv crossings.



Figure 5: Scan order is kept when passing from  $(\Theta, R)$  to (x, z). This is not the case when passing from  $(\tau, R)$  to (x, y).



Figure 6: The points are accumulated in the pixels of the facade grid. The accumulation volume of a given pixel is shown in red. It extends to infinity behind the grid.

#### 2.2 Prismatic coordinate system

We propose a new way to accumulate the points in the facade grid. The idea is to use a virtual sensor grid. For each pixel of the facade grid, there is a corresponding pixel on the sensor grid. Ideally, the following property is ensured: any laser beam passing through a facade pixel passes through the corresponding sensor pixel (fig: 6). The virtual sensor grid is positioned in order to approach this ideal.

The function that associates prismatic coordinates (u, v, h) to the space coordinates (x, y, z) is explained in the 2d and 3d cases. Then a method to place the grid is proposed.



Figure 7: 2d formulation

**2d formulation.** We place ourselves in the horizontal plane (fig:7). We look for the (u, h) coordinates of a point P(x, y). [AA'] symbolizes the sensor location, and [BB'] the facade footprint. The laser beam is approximated by a ray [ab) that passes through P and intersects (AA') in a and (BB') in b, such that

$$\overrightarrow{Aa} = u\overrightarrow{AA'}$$
 and  $\overrightarrow{Bb} = u\overrightarrow{BB'}$  (2)

The straight line (ab) is thus the iso-u of P.

The position of P on [ab) is given by h. We propose to use the following equation that sets h as the signed distance to b normalized by |ab|, b being the virtual intersection of the facade plane and the laser beam.

$$h = \overrightarrow{bP} \cdot \frac{\overrightarrow{ba}}{|ba|^2} \tag{3}$$

We will show how to find u for a given point P in order to obtain the couple (u, h). In this context, P, A, A', B and B' are known. As P lies on (ab), we have:  $\overrightarrow{aP} \times \overrightarrow{ab} = 0$  which is equivalent to:

 $\begin{array}{l} u^2(\overrightarrow{BB'}\times\overrightarrow{AA'})+u(\overrightarrow{AP}\times\overrightarrow{BB'}-\overrightarrow{BP}\times\overrightarrow{AA'})+\overrightarrow{AP}\times\overrightarrow{AB}=0 \\ (4) \\ \text{So we obtain a quadratic equation of the form } \alpha u^2+\beta \, u+\gamma=0 \\ \text{that allows two solutions for } u. \text{ Note that if } \overrightarrow{AA'} \text{ and } \overrightarrow{BB'} \text{ are colinear, } \alpha=\overrightarrow{BB'}\times\overrightarrow{AA'}=0, \text{ and the equation becomes } \beta u+\gamma=0 \Leftrightarrow u=-\frac{\gamma}{\beta}, \text{ giving us the unique solution} \\ \text{for } u. \text{ In the general case } (\overrightarrow{AA'} \text{ and } \overrightarrow{BB'} \text{ not colinear}), \text{ one solution corresponds to a degenerate case, that brings } a \text{ and } b \text{ far from } P. \text{ One can verify that, defining } q=-\frac{\beta+sign(\beta)\sqrt{\Delta}}{2} \\ \text{and } \Delta=\beta^2-4\alpha\gamma, \text{ the two roots of the quadratic equation are } u_d=\frac{q}{\alpha} \text{ and } u_s=\frac{\gamma}{q}. \text{ As } \lim_{\alpha\to 0}u_d=\pm\infty \text{ and } \lim_{\alpha\to 0}u_s=-\frac{\gamma}{\beta}. \\ u_d \text{ is a degenerate solution and } u_s, \text{ the searched value for } u. \end{array}$ 

#### Special cases.

If AA' and BB' are colinear, and ||AA'|| = ||BB'||, b is the orthogonal projection of P on (BB').

If  $|AA'| \to 0$ , we obtain a kind of angular projection.



Figure 8: 3d formulation

**3d formulation.** As in figure 8,  $\overrightarrow{AA'} \perp \overrightarrow{AA''}$  and  $\overrightarrow{BB'} \perp \overrightarrow{BB''}$ . For our application, we assume that  $\overrightarrow{AA''}$  and  $\overrightarrow{BB''}$  are colinear. u and v are the horizontal and vertical coordinates of P in both the basis  $\{AA', AA''\}$  and  $\{BB', BB''\}$ . h is the signed distance to b.

$$\overrightarrow{Aa} = u\overrightarrow{AA'} + v\overrightarrow{AA''}$$
$$\overrightarrow{Bb} = u\overrightarrow{BB'} + v\overrightarrow{BB''} \quad \text{and} \quad h = \overrightarrow{bP} \cdot \frac{\overrightarrow{ba}}{|ba|}$$

**Placing the grids.** The rectangle with the B, B' and B'' corners is placed along the facade wall. The rectangle with the A,

A' and A'' corners is placed along the trajectory in order to align laser beams with iso-uv. For this purpose, we only adapt the horizontal position of the rectangle. It is vertical with an height close to zero because the location of the sensor is not supposed to move vertically.  $\overrightarrow{AA'}$  is fitted with RANSAC, by selecting randomly pairs of laser beams. Each pair of laser beams is a potential couple AB, A'B' defining a prismatic coordinate system. The score of each pair must reflect the alignment between the laser beams and the iso-uv. The score is thus the sum of the dot products between the laser beam and the iso-uv of each 3d point.

# 2.3 Comparison between Cartesian and prismatic coordinate systems

The orthogonal and prismatic coordinate systems are compared through some examples in figures 9, 10, 11 and 12. Many problems in pixel accumulation and surface reconstruction can be avoided using a "sensor-oriented" coordinate system such as the prismatic one.



Figure 9: The facade in profile and a wall light are sketched. The points are accumulated on the vertical plane, according to an orthogonal projection (a), (b) and a prismatic projection (c), (d). Orthogonal projection leads to accumulating points from distant beams in a same pixel (a), causing an ambiguity in the choice of the surface (b). prismatic projection is more consistent with the scan (c) and leads to a surface reconstruction with less ambiguity (d).



Figure 10: Laser beams (a). Surface reconstructed in a Cartesian coordinate system (b) and a prismatic coordinate system (c). The use of Cartesian coordinate system may leads to a surface that "goes through" the wall because it is attracted by points of the room ceiling behind the facade (1). The surface is also attracted to a wall light (2). The surface reconstructed using the prismatic coordinate system follows the beam trajectory, it goes trough the window (1) and wraps the wall light (2).



Figure 11: Loss of Accuracy

(a) Laser beams partially entering a cavity.

(b) The iso-*uv* (purple) are along the laser beams, points are distributed homogeneously in the pixels. This is the best configuration for surface reconstruction: there is one point per pixel, there is no ambiguity in depth computation.

(c) Orthogonal projection, iso-uv (pink) orthogonal to the wall. Three points are accumulated in the same pixel (2) while there is no point in some other pixels (1).



# Figure 12: Inconsistency between the surface and the laser beams

(a) An object hides a part of the wall behind.

(b) Prismatic coordinate system: The reconstructed surface does not fit everywhere with the actual surfaces, but is not pierced by the laser beams.

(c) Cartesian coordinate system: The reconstructed surface is intersected by two laser beams (1).

# **3 DEFORMABLE GRID**

# 3.1 Iterative deformation of the grid

We solve the problem of finding an optimal depth h for each grid pixel as an energy minimization with a data term that shifts h to the primitives, and a smoothness term that retains h close to the neighboring pixels. Each iteration, the depth of each pixel is moved from  $h_n$  to  $h_{n+1}$ .



 $h_n$ : previous h

 $h_{\text{pixel}}$ : h estimated from primitives accumulated in the pixel.  $h_{\text{neigh}}$ : h estimated from the neighboring pixels.

## 3.2 Strategy for pixel depth computation

The strategy for  $h_{\text{pixel}}$  and  $h_{\text{neigh}}$  calculation may vary according to the data or the desired output. In our case, we use a **bilateral filter** that allows discontinuities like at window edges, but smoothes the wall flat surfaces. Pixel depth is computed thanks to the following equations, where  $\exp(-\frac{x^2}{2\sigma^2})$  is denoted by  $G(\sigma, x)$ .

$$h_{\text{pixel}} = \frac{\sum G(\sigma_h, h_i)h_i}{\sum G(\sigma_h, h_i)}$$
(6)

Where  $h_i$  is the depth value of the  $i^{th}$  primitive accumulated in the pixel. The Gaussian function weights the contribution of each  $h_i$ . If an  $h_i$  is far from the current depth, its contribution is low. This allows a relative independence to the outliers. The smoothness term is provided by the depth values of the eight neighbor pixels. Hence, the neighbors depth is given by:

$$h_{\text{neighbors}} = \frac{\sum_{i=1}^{8} G(\sigma_h, h_i) G(\sigma_d, d_i) h_i}{\sum_{i=1}^{8} G(\sigma_h, h_i) G(\sigma_d, d_i)}$$
(7)

 $h_i$  is the  $h_{\text{pixel}}$  value of the  $i^{th}$  neighbor pixel.  $d_i$  is the distance between the pixel center (u, v) and the  $i^{th}$  neighbor pixel center  $(u_i, v_i)$ . If the depth of a neighbor pixel is far from the current pixel depth, its contribution is low, allowing a discontinuity at the edge between these pixels. The values of  $\sigma_h$  and  $\sigma_d$  have to be fixed, one can choose the grid accuracy that is, the length of a pixel side.

Some other strategies are possible. For instance, the closest point from the facade plane can be kept. The median would give a robust estimation if there are many point. Other quantile values would be used if the closest or furthest points are looked for.

# 3.3 Primitives

**Points / Lidar echos:** If many pixels are empty of 3d points, the iterative process may be slow to converge. A solution is to start with a low resolution grid (large pixels). This allows to initialize a more precise grid. This procedure can be repeated many times, the same way a scale-space pyramid is built.

**Triangles:** The advantage of the triangles over the points is that they provide surfaces. This allows the algorithm to converge more rapidly. Thus, the triangle soup provided by sensor geometry can be used as input.

**Line segments:** Line segments extracted from scan lines can be accumulated in grid pixels. This option is mentioned because many papers perform processes such as classification on scanlines (Hadjiliadis and Stamos, 2010). simplified scanlines are more compact than points and their projections could intersect more pixels.

**Mixed primitives:** Nothing prevents to accumulate heterogeneous primitives.



Figure 13: Grid constructed for a facade (left). Detail of the grid (right).

We display some visual results, showing the precision of the proposed method, despite the low point density (fig: 13). Parameter adjustments permits a compromise between a smooth surface as in figure 14 (a), and more rugged and closer to the points, 14 (c).



Figure 14: Three parameterizations for the same facade piece (three-quarters and profile view). From left to right, the data term increases and the smoothness term decreases. The surface becomes less simple and more rugged, but closer to the lidar points. (a)  $\lambda_{pixel} = 0.01$  and  $\lambda_{neigh} = 0.99$ (b)  $\lambda_{pixel} = 0.10$  and  $\lambda_{neigh} = 0.90$ 

(c)  $\lambda_{\text{pixel}} = 0.90$  and  $\lambda_{\text{neigh}} = 0.10$ 

# 5 CONCLUSION AND FUTURE WORK

We presented an approach to reconstruct facade geometry from lidar data. We aimed at surfaces consistent with the data, especially the optical images acquired in the same time. The proposed framework allows to reconstruct 2.5d grids in a coordinate system adapted to the sensor point of view. For this purpose, we introduced a prismatic projection that is a generalization of the orthogonal projection and the corresponding coordinate system. It allows to benefit from the sensor geometry while providing a coordinate system topologically consistent with the 3d space. The output 2.5d grids are suitable for immersive navigation and other applications that need compact and detailed 3d models. They could also be used as inputs for facade structure analysis, in particular for window detection or grammar rules extraction.

In future work, we would like to texture the grid with the optical images. For this task, our framework should allow us to deal with images, that are also 2.5d grids in image coordinate system, and to project image pixels onto the reconstructed grid.

#### References

- Badino, H., Huber, D., Park, Y. and Kanade, T., 2011. Fast and accurate computation of surface normals from range images. In: ICRA, pp. 3084–3091.
- Becker, S. and Haala, N., 2009. Grammar supported facade reconstruction from mobile lidar mapping. ISPRS.
- Chauve, A.-L., Labatut, P. and Pons, J.-P., 2010. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In: CVPR, pp. 1261–1268.

- Demantké, J., Vallet, B. and Paparoditis, N., 2012. Streamed vertical rectangle detection in terrestrial laser scans for facade database production. ISPRS Annals I-3, pp. 99–104.
- Deschaud, J.-E., 2010. Dense point cloud processing and 3D environment modeling from LiDAR/Camera mobile system. PhD thesis, Ecole Nationale Supérieure des Mines de Paris.
- Friedman, S. and Stamos, I., 2011. Real time detection of repeated structures in point clouds of urban scenes. In: 3DIM-PVT, pp. 220–227.
- Frueh, C., Jain, S. and Zakhor, A., 2005. Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. IJCV 61(2), pp. 159–184.
- Furukawa, Y., Curless, B., Seitz, S. M. and Szeliski, R., 2009. Reconstructing building interiors from images. In: ICCV, pp. 80– 87.
- Hadjiliadis, O. and Stamos, I., 2010. Sequential classification in point clouds of urban scenes. In: 3DPVT.
- Hadsell, R., Bagnell, J. A. and Hebert, M., 2009. Accurate rough terrain estimation with space-carving kernels. In: RSS.
- Lafarge, F., Keriven, R., Brédif, M. and Vu, H., 2013. A hybrid multi-view stereo algorithm for modeling urban scenes. TPAMI.
- Li, Y., Zheng, Q., Sharf, A., Cohen-Or, D., Chen, B. and Mitra, N. J., 2011. 2d-3d fusion for layer decomposition of urban facades. In: ICCV, pp. 882–889.
- Müller, P., Zeng, G., Wonka, P. and Van Gool, L., 2007. Imagebased procedural modeling of facades. TOG 26(3), pp. 85.
- Musialski, P., Wimmer, M. and Wonka, P., 2012a. Interactive coherence-based façade modeling. In: Computer Graphics Forum, Vol. 31number 2pt3, pp. 661–670.
- Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., van Gool, L. and Purgathofer, W., 2012b. A survey of urban reconstruction. Eurographics, State of the art reports.
- Nan, L., Sharf, A., Zhang, H., Cohen-Or, D. and Chen, B., 2010. Smartboxes for interactive urban reconstruction. TOG 29(4), pp. 93.
- Paparoditis, N., Papelard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N. and Houzay, E., 2012. Stereopolis II: A multi-purpose and multi-sensor 3d mobile mapping system for street visualisation and 3d metrology. RFPT 200, pp. 69–79.
- Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H. and Guibas, L. J., 2008. Discovering structural regularity in 3d geometry. In: TOG, Vol. 27number 3, p. 43.
- Pu, S. and Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. In: ISPRS, LS.
- Shalom, S., Shamir, A., Zhang, H. and Cohen-Or, D., 2010. Cone carving for surface reconstruction. TOG 29(6), pp. 150.
- Shen, C.-H., Huang, S.-S., Fu, H. and Hu, S.-M., 2011. Adaptive partitioning of urban facades. In: TOG, Vol. 30number 6, p. 184.
- Wang, R., Bach, J. and Ferrie, F. P., 2011. Window detection from mobile lidar data. In: WACV, pp. 58–65.
- Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N. J., Cohen-Or, D. and Chen, B., 2010. Non-local scan consolidation for 3d urban scenes. TOG 29(4), pp. 94.