Depth Map based Facade Abstraction from Noisy Multi-View Stereo Point Clouds

Andreas Ley and Olaf Hellwich

Computer Vision & Remote Sensing Group, TU Berlin, Berlin, Germany
{andreas.ley,olaf.hellwich}@tu-berlin.de

Abstract. Multi-View Stereo offers an affordable and flexible method for the acquisition of 3D point clouds. However, these point clouds are prone to errors and missing regions. In addition, an abstraction in the form of a simple mesh capturing the essence of the surface is usually preferred over the raw point cloud measurement. We present a fully automatic pipeline that computes such a mesh from the noisy point cloud of a building facade. We leverage prior work on casting the computation of a 2.5D depth map as a labeling problem and show that this formulation has great potential as an intermediate representation in the context of building facade reconstruction.

1 Introduction and Related Work

Multi-View Stereo (MVS) has become a viable tool for the 3D reconstruction of objects. In the context of 3D city modeling, the relative affordability and ease of use makes MVS an interesting choice for the reconstruction of building facades. Due to lack of texture, however, the point clouds produced by MVS often contain holes and outliers. In addition to those issues, a form of abstraction is usually desired which removes clutter and noise as well as irrelevant details. Such an abstraction provides the essence of the surface, usually in polygonal form rather than as points or voxels.

Both aspects, noise removal and abstraction, require strong priors as well as a geometric representation of the facade that supports them. We focus on the use of depth maps cast as Markov Random Fields for cleaning as well as abstracting building facade reconstructions. Based on this concept, we demonstrate a fully automatic pipeline, point cloud to mesh (see Figure 1), which offers many opportunities for additional semantic priors. Our approach is heavily inspired by prior work which we summarize in this section. We describe the individual steps of our processing chain in Section 2 and show results and comparisons in Section 3. Additional results can be found in the supplementary material. We close with a conclusion and discussion of future work in Section 4.

The idea to use MVS methods in conjunction with strong priors for building facade reconstruction is by no means new. The prior work in this field is too extensive to cover it here in its entirety. Instead we discuss a representative subset to sketch the extend of the different approaches.

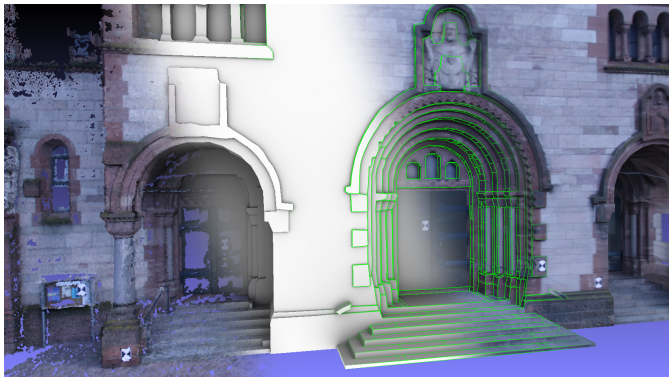

Fig. 1. We present a method to compute simple but concise meshes (shown with ambient occlusion in the middle and textured on the right) from noisy Multi-View Stereo point clouds (left) of building facades by using a depth map as an intermediate representation and using a Markov Random Field based formulation to incorporate strong regularization. Green lines, where visible, highlight geometric edges between planar regions. Based on the Herz-Jesu-P25 dataset of [15].

Works explicitly targeting building facades range in their expressive power from ruled vertical surfaces [5] over flat facades with parameterized windows and doors [18] to arbitrary but axis aligned and facade-parallel rectangles [19]. We lend the idea of using an orthographic depth map from [19], but are less restrictive in our choice of further regularization, allowing shapes other than rectangles. Similarly, we are not limited to a set of handcrafted shapes as in [18].

When extending the scope from building facades to man made structures in general, the wealth of explored regularizations is even more extensive. A recurring theme, however, is the use of Markov Random Field (MRF) formulations. The methods in [6, 14] compute a dense depth map for one of the input images by assigning labels to pixels. Each label corresponds to a plane which is aligned to one of the three main directions in man made structures [6] or derived from intersecting lines [14]. To promote large, homogeneous areas, a smoothness constraint is used in both cases. We adopt the MRF formulation but apply it not in the image space of one of the input images but instead in a coordinate system embedded into the facade. We also restrict the planes' orientations to be parallel to the facade. This is similar to the modeling of wall details in [10] but without the rank reduction which enforces axis aligned rectangular shapes.

The image space depth map of [6] is extended to a full 3D model in [7] by fusing the individual depth maps. Another full 3D representation is proposed in [3] which splits the space into convex volumes based on extracted planes and then labels the volumes as solid or free based on a graph cut formulation. The expressive power of both cases is obviously higher than in our 2.5D approach. For most building facades, however, a 2.5D representation is sufficient if not even better suited.

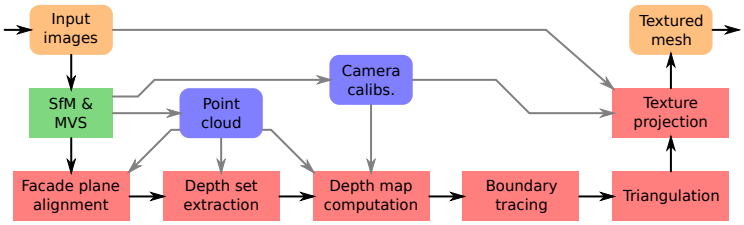

Fig. 2. Overview of the proposed processing chain. Orange: Input and output. Green: Stock SfM/MVS pipeline which produces the input to our method (blue). Red: Our method. Black arrow: Processing flow. Gray arrows: Additional data flow.

2 Methodology

Our approach is summarized in Figure 2. In the default case, the only input are images of the facade. We use standard SfM and MVS [8] pipelines to estimate the camera parameters and the point cloud of the surface. Not unlike [19] and many others, we use a depth map as an intermediate representation even though the final result is a mesh.

Using a depth map requires an accurate estimation of the facade's orientation (see Section 2.1). This includes the orientation of the facade plane as well as an appropriate in plane rotation to align horizontal and vertical directions with the axes of the depth map. The depth map is filled with depth values by first compiling a set of discrete depth hypotheses and then labeling the pixels of the depth map with those hypotheses (see Section 2.2). The depth map is converted back into a mesh by tracing the boundaries between depth segments and performing a constrained Delaunay triangulation (see Section 2.3). Finally a texture map is computed from the input images (see Section 2.4).

2.1 Plane Alignment

The estimation of the depth map as well as future additional semantic processing require an accurate estimation of the facade orientation. In addition, a rotation within the facade plane needs to be estimated to align horizontal and vertical directions in the facade with the coordinate axes of the depth map. Figure 3 summarizes our facade orientation estimation approach.

We run a modified version of PEaRL [11] to fit planes to the point cloud. PEaRL alternates between a labeling phase and a model fitting phase. In the labeling phase, data points are assigned to models (planes) under a smoothness constraint that promotes equal labels for neighboring data points. In the model fitting phase, the models (planes) are readjusted to fit the corresponding data points as closely as possible. To circumvent the problem of defining neighborhoods in a point cloud, we replace the MRF based alpha expansion of the labeling phase in PEaRL with the dense conditional random field (dCRF)

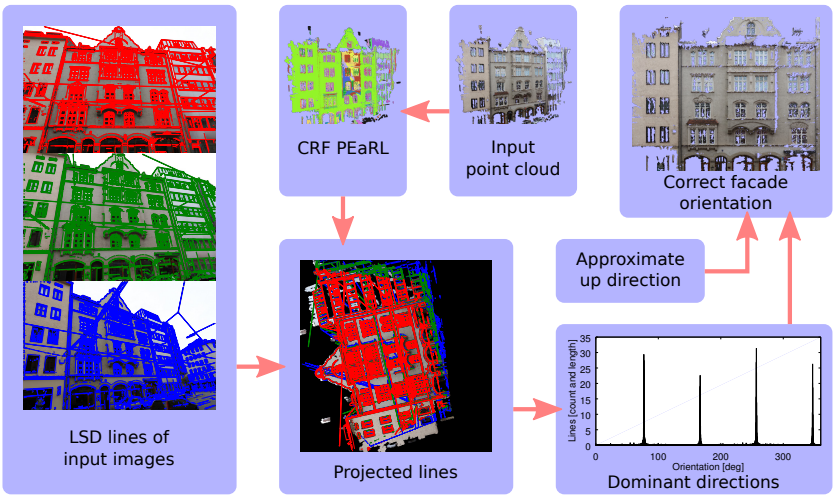

Fig. 3. Automatic computation of the facade orientation. See text for details.

formulation of [12]. A dCRF simply connects all nodes with each other with weights computed from Gaussian kernels applied to the nodes' feature vectors. Similarly to [12] we use position and color as features, with the minor difference that our points have 3D positions.

The plane with the most support is extracted as the main facade plane. It usually fits the facade very closely as the points in windows and balconies are explained away with secondary planes. Nonetheless, we use an additional clutter model in PEaRL to soak up spurious outliers.

With the facade plane at hand, the facade still needs to be rotated such that the roof points up in the depth map. This is beneficial to the grid structure of the depth map and will facilitate future semantic analysis. We make the common assumption that for man made structures such as facades, most lines are either horizontal or vertical and can be used to align the facade rotation. We extract LSD line segments [9] from the input images and project them onto the facade plane. Computing the histogram of line directions yields four very clear spikes from which the facade rotation can be judged up to an ambiguity of multiples of 90° (see Figure 3). An approximate up direction from each input image in the form of the average blue gradient is used to resolve this ambiguity. Another possibility is to use the orientation meta data in the jpeg exiv tags which might provide a more reliable hint.

2.2 Orthographic Depth Map

Since the facades that we are interested in are mostly flat, we adopt the assumption laid out in [19] that the facade geometry can be described in a 2.5D fashion by the depth map of an orthographic projection of the facade. This description

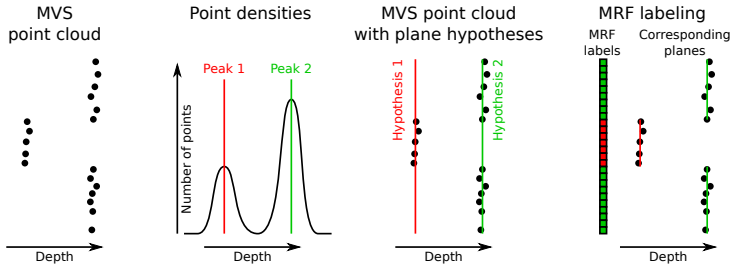

Fig. 4. Overview of the MRF approach to the depth map computation. From left to right: 1) Point cloud projected into the coordinate system of the facade. 2) Histogram of occurring depths. 3) Depth hypotheses extracted from the histogram. 4) MRF based labeling of the depth map pixels, each label of which corresponds to a depth.

has the elegant property that surfaces parallel to the facade plane, which are well exposed to the camera and thus well supported by the data, are explicitly modeled while the surfaces perpendicular to the facade plane, which are often missing in the data, are modeled implicitly by depth jumps in the depth maps.

We embed the depth map in the coordinate system of the facade and limit it horizontally and vertically to the extend of the central 98% of the input points on the facade plane. The depth map must be filled with depth values based on the input data as well as possible regularizations. In [19], the (primary) regularization is to enforce rectangular regions of equal depth. The same prior, although realized by different means, can be found in [10] as well where it is applied to wall details inside rooms. Since most surfaces of a building facade are either perpendicular or parallel to the main facade plane, this restriction to regions of homogeneous depth performs quite well in practice. The rectangular outlines however, while working great with most windows and doors, are too restrictive for our purposes as we seek to reconstruct curved structures such as arched windows as well.

Instead we adapt the pixel labeling approach of [6, 14] to operate directly within the orthographic projection. We project the input point cloud into the facade-aligned coordinate system and compute a histogram of depth values (see Figure 4). A limited set of depth hypotheses is then extracted from this histogram. They represent possible depths that the pixels of the depth map can adopt. This step is very similar to [6] except that we limit the depth hypotheses to one axis instead of all three. The actual filling of the depth map with depth values is cast as a labeling problem of the depth map pixels where the individual labels are the depth hypotheses. Just like [6], we formulate this problem as the minimization of the following energy:

$$E = \sum_{p} E_d(h_p) + \sum_{p,q \in \mathcal{N}(p)} E_s(h_p, h_q) \; . \tag{1}$$

The unary data term $E_d(h_p)$ represents the cost of assigning a specific hypothesis h_p to depth map pixel p and the binary smoothness term $E_s(h_p, h_q)$ models

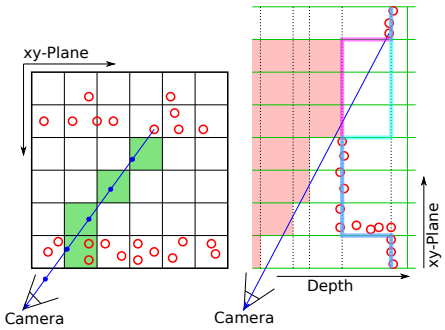

Fig. 5. Accumulation of free space votes. For each point, the depth map pixels towards all its observing cameras are considered in turn (left side) and all hypotheses that would block the view ray recieve a data term penalty for that pixel (right side).

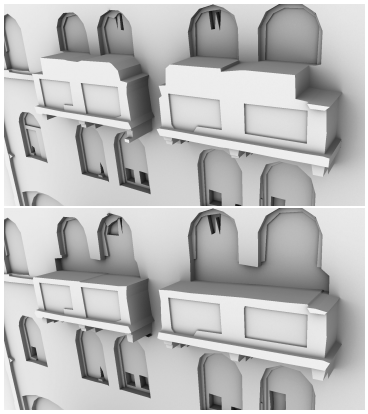

Fig. 6. Impact of the free space votes on occluded regions. Top without and bottom with free space votes. The two cases correspond to the pink and blue lines on the right side of Figure 5.

the cost of assigning (possibly differing) hypotheses h_p and h_q to neighboring pixels p and q. When interpreted as a Markov Random Field (MRF), the solution to this energy minimization can be approximated through repeated graph cuts via alpha expansion [2].

The cost of data term $E_d(h_p)$ is computed from a) the distances between the depth hypothesis h_p and the point cloud points that fall onto pixel p and b) the number of free space votes. The free space votes represent the knowledge that the view rays between point cloud points and their observing cameras must be unobstructed (amongst others see [6, 14]). We accumulate those votes up front by walking the path from each point cloud point to all of its observing cameras through the depth map. For each covered pixel, votes are accumulated for all hypotheses that would occlude the view ray (see Figure 5). We found this to be extremely helpful in resolving ambiguities at occlusion boundaries like balconies (see Figure 6).

The cost of the smoothness term $E_s(h_p, h_q)$ is zero for equal labels $h_p = h_q$ and always greater than zero for differing labels to enforce large, uniform depth regions. Similarly to [6], we assume that geometric edges often coincide with visual edges in the images and thus modulate the cost with the strength of edge gradients in the input images. Contrary to the image space based depth map in [6], where a direct mapping between depth map locations and image locations exists, our depth map does not live in the same coordinate system as one of the input images. Instead the depths of the depth hypotheses and the camera calibrations are used to find the correct location of the front and back edge of a depth jump in the input images.

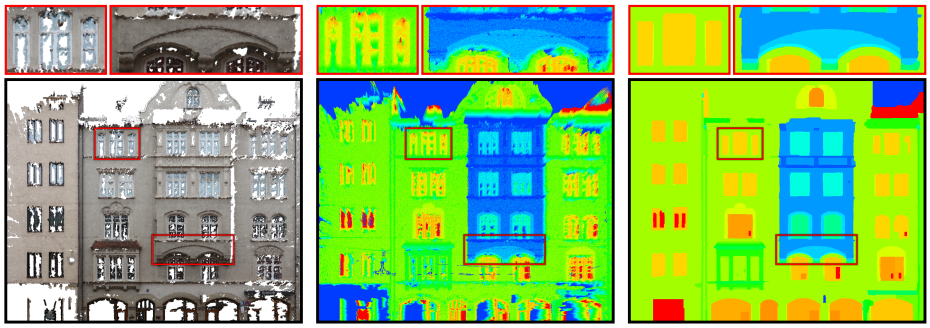

Fig. 7. Left to right: Point cloud (for reference), initial depths (based solely on $E_d(h_p)$), final depths.

Figure 7 shows the results that can be achieved with the described approach. Note that the regularization removes most of the outliers while retaining to a large degree the (subjectively) important structures such as windows, balconies, columns, and arches. Also note that a reasonable, albeit primitive inpainting of holes in the point cloud is achieved as well. Contrary to the approaches outlined in [19, 10], we are not restricted to axis aligned rectangles and thus recover arches and round windows as well, although at the cost of more "wobbly" windows.

2.3 Meshing

After the computation of the depth map, we seek to create a low poly mesh of the facade. We first trace and simplify the boundary curves between the depth segments with an approach similar to Variational Shape Approximation (VSA) [4] only operating on lines instead of triangles.

The boundaries between differing depth labels in the depth map are cast as a graph of connected line segments. Straight lines are fitted to larger, junction free stretches of line segments by alternating between model assignment and model fitting. During model assignment, line segments are assigned to one of the straight lines (models) through a greedy region/line growing process along the graph starting from seeds. During the model fitting, the straight lines (models) are adjusted to fit as closely as possible to the assigned line segments. Additional straight lines (models) are added and optimization is performed until the set of straight lines approximates the underlying depth boundaries sufficiently. Anchor vertices are placed automatically between connected straight lines and at junctions, yielding simplified and smoothed segment boundaries. Figure 8 shows the result of this contour extraction.

With the boundary curves represented as a set of connected (anchor) vertices, the individual facade elements are meshed with a constrained Delaunay triangulation. Finally, the faces perpendicular to the facade plane are added.

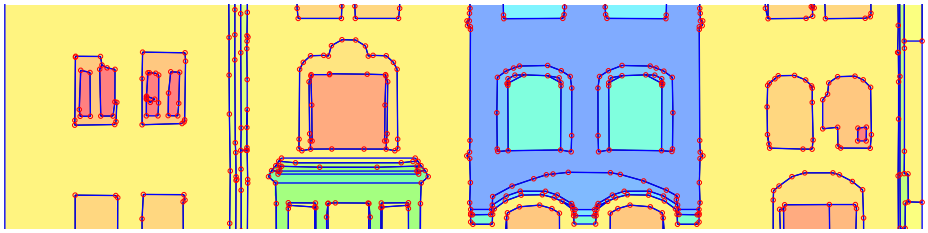

Fig. 8. Simplified contour lines. Notice that the simplified contour lines smooth out the round boundaries of the arches while retaining sharp window corners.

2.4 Texturing

Finally, we create a texture by projecting the input images onto the mesh. Since we do not want the balconies to bleed onto the facade, or the facade to bleed onto the windows, occlusions have to be taken into account. We use the pose of the facade as well as the camera parameters to render a depth map for each input image which contains in each pixel the distance of the closest intersection with the surface along the corresponding view ray. This map is easily obtained through rasterization of the mesh in the input image space.

The 3D mesh is unfolded into the 2D texture space using the automatic UV layouting tool in Blender [1]. Using those 2D texture coordinates, we then rasterize the mesh into the texture. For each texel and each camera, the depth in the camera's view space is computed and compared to the depth stored in the camera's depth map to determine if the camera actually observes that part of the surface or whether it is occluded. In the unoccluded case, the color is sampled from the input image and the colors from all (unoccluded) cameras are averaged and stored in the texture. Occlusions from objects not present in the mesh, such as street lights, land lines, and passing cars, are presently not handled.

3 Evaluation

Abstractions, as in reduction to the essence, always contain a certain subjective and artistic aspect. This makes quantitative evaluations rather complicated and the method presented here is no exception. For reference we show the precision of the produced mesh in Figure 9 alongside the precision of the input point cloud. The images and reference data are the Herz-Jesu-P8 dataset of [15]. As expected, the abstraction results in an increased error.

Nonetheless, in less forgiving datasets with more outliers, our approach compares favorably to a baseline poisson mesh of the input point cloud (see Figure 10). Some remaining errors are visible in the right cut-out of Figure 10 which can be further suppressed by additional constraints like symmetry.

Depending on the actual use case, however, those remaining errors might actually be tolerable or hidden when a texture is applied to the mesh. For point

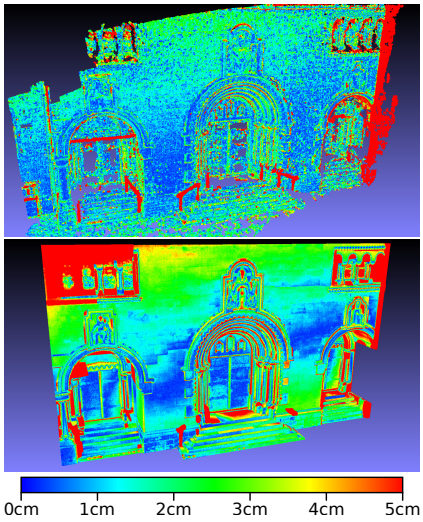

Fig. 9. Precision as distance from reference data, color coded from zero to five centimeters as blue through red. Top: PMVS point cloud. Bottom: Ours.

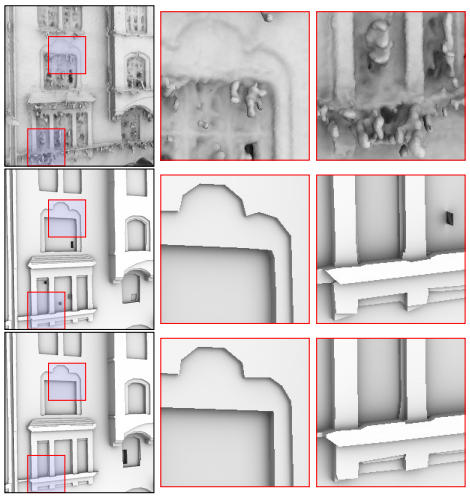

Fig. 10. Comparison with a baseline poisson mesh. Top to bottom: Poisson, ours, ours with manually specified symmetries (additional links in the MRF).

clouds with gross errors as well as missing regions, Figure 11 demonstrates aesthetically pleasing meshes. Note that essential structures like balconies, windows, doors, rain pipes, etc. are reconstructed well while otherwise keeping the geometry to a minimum. Additional reconstructions can be found in the supplementary material.

4 Conclusion and Future Work

We leverage the prior work of casting dense depth map estimation as a labeling problem and apply it to building facade reconstruction. It performs well in this context and presents a multitude of options for further regularization. A method to extract a simplified mesh from this depth map is described as well, which makes the depth map a viable intermediate representation even if a mesh is desired as the final result.

We demonstrate our method on noisy and incomplete MVS point clouds and compare it to a baseline poisson mesh. The proposed approach creates visually pleasing abstractions of facades, retaining the major geometric entities while removing clutter and noise alike.

In the future we seek to experiment with various additional priors, such as automatic detection of symmetries and repetitions. Inferring such semantic data, usually supported by shape grammars, has been demonstrated for 2D and 3D input (for example, see [16] and [13] respectively). Semantic input in the

Fig. 11. Top: Input point cloud. Bottom: Textured mesh with highlighted seams.

form of detected windows can help remove regions where the MVS pipeline reconstructed the curtains or other interior surfaces. Such constraints must be included in the energy formulation of Equation (1). The data and smoothness terms provide a wealth of opportunities in that regard. The smoothness cost can be reduced along lines that should be favored for depth jumps. This can be used to guide edges along dominant lines or to restrict the borders of semantic objects. The data cost can also be modified in favor of certain labels in specific depth map regions, for example to favor a certain depth for windows. To encourage symmetries, additional links can be added to the MRF formulation between pixels that should have the same label (see bottom of Figure 10 for an example).

Right now, the only "primitive type" are facade parallel planes. However, sloped planes as in [14] or axis aligned cylinders are also conceivable within this framework. Alternating between pixel labeling and readjusting the hypotheses in a fashion similar to PEaRL [11] might further increase the precision. Proper sky detection as in [19] is another avenue worth exploring as it can provide an upper boundary for the facade.

Even though texturing is not the central focus of our attention, some simple improvements are of interest. Street lights and cars sometimes leave faint ghost images on the facades since they are not geometrically reconstructed and thus not considered in the occlusion handling. A quite advanced solution is presented in [17] which could be integrated in the future.

Acknowledgements. This paper was supported by a grant (HE 2459/21-1) from the Deutsche Forschungsgemeinschaft (DFG). The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-45886-1_13

References

1. Blender, http://www.blender.org
2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)
3. Chauve, A.L., Labatut, P., Pons, J.P.: Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In: CVPR. pp. 1261–1268 (2010)
4. Cohen-Steiner, D., Alliez, P., Desbrun, M.: Variational shape approximation. In: ACM SIGGRAPH 2004 Papers. pp. 905–914. SIGGRAPH '04 (2004)
5. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.: 3d urban scene modeling integrating recognition and reconstruction. International Journal of Computer Vision 78(2), 121–141 (2007)
6. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Manhattan-world stereo. In: CVPR. pp. 1422–1429 (2009)
7. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Reconstructing building interiors from images. In: ICCV. pp. 80–87 (2009)
8. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. IEEE Trans. on Pattern Analysis and Machine Intelligence 32(8), 1362–1376 (2010)
9. Grompone von Gioi, R., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a Line Segment Detector. Image Processing On Line 2, 35–55 (2012)
10. Ikehata, S., Yang, H., Furukawa, Y.: Structured indoor modeling. In: ICCV. pp. 1323–1331 (2015)
11. Isack, H., Boykov, Y.: Energy-based geometric multi-model fitting. International Journal of Computer Vision 97, 123–147 (2012)
12. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 24, pp. 109–117. Curran Associates, Inc. (2011)
13. Martinovic, A., Knopp, J., Riemenschneider, H., Van Gool, L.: 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. In: CVPR. pp. 4456–4465 (2015)
14. Sinha, S.N., Steedly, D., Szeliski, R.: Piecewise planar stereo for image-based rendering. In: ICCV. pp. 1881–1888 (2009)
15. Strecha, C., von Hansen, W., Gool, L.J.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR (2008)
16. Teboul, O., Kokkinos, I., Simon, L., Koutsourakis, P., Paragios, N.: Parsing facades with shape grammars and reinforcement learning. IEEE Trans. on Pattern Analysis and Machine Intelligence 35(7), 1744–1756 (2013)
17. Waechter, M., Moehrle, N., Goesele, M.: Let there be color! Large-scale texturing of 3D reconstructions. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV. Lecture Notes in Computer Science, vol. 8693, pp. 836–850. Springer (2014)
18. Werner, T., Zisserman, A.: Model selection for automated architectural reconstruction from multiple views. In: Proceedings of the British Machine Vision Conference. pp. 53–62 (2002)
19. Xiao, J., Fang, T., Zhao, P., Lhuillier, M., Quan, L.: Image-based street-side city modeling. In: ACM SIGGRAPH Asia 2009 Papers. pp. 114:1–114:12. SIGGRAPH Asia '09 (2009)