CEIG - Spanish Computer Graphics Conference (2018)
I. García-Fernández and C. Ureña (Editors)

Depth map repairing for building reconstruction

C. Andujar[1], O. Argudo[1], I. Besora[2], P. Brunet[1], A. Chica[1], M. Comino[1]

[1]VirVIG, Computer Science Department, Universitat Politecnica de Catalunya
[2]Institut Cartogràfic i Geològic de Catalunya

Abstract

Structure-from-motion along with multi-view stereo techniques jointly allow for the inexpensive scanning of 3D objects (e.g. buildings) using just a collection of images taken from commodity cameras. Despite major advances in these fields, a major limitation of dense reconstruction algorithms is that correct depth/normal values are not recovered on specular surfaces (e.g. windows) and parts lacking image features (e.g. flat, textureless parts of the facade). Since these reflective properties are inherent to the surface being acquired, images from different viewpoints hardly contribute to solve this problem. In this paper we present a simple method for detecting, classifying and filling non-valid data regions in depth maps produced by dense stereo algorithms. Triangles meshes reconstructed from our repaired depth maps exhibit much higher quality than those produced by state-of-the-art reconstruction algorithms like Screened Poisson-based techniques.

CCS Concepts

•***Computing methodologies*** → ***Computer graphics;***

1. Introduction

The construction of detailed 3D models of existing buildings has a number of applications in areas such as videogames, architecture and smart cities. Although 3D scanner equipment (e.g. Lidar) can produce detailed and accurate models of 3D buildings, techniques based on bare images taken from commodity digital cameras are gaining popularity due to their low cost and general availability. In this case, the most common approach is to combine Structure from Motion (SfM) and Multi-view stereo (MVS) techniques (Fig. 1).

SfM [SF16] takes as input a collection of images of an object (e.g. a building), taken from different viewpoints, and outputs both camera parameters (both intrinsic and extrinsic) for the images, and a sparse point cloud representation of the scene. MVS [SZPF16] takes as input the provided images and the reconstructed camera poses, and outputs both undistorted versions of the images, and a dense reconstruction in the form of depth maps and color maps for each input image. These depth maps can be fused into a single colored and oriented point cloud, which can be further meshed through surface reconstruction algorithms such as Screened Poisson-based reconstruction [KH13]. Further processing allows for the generation of color and normal texture atlases for higher normal and color resolution on the reconstructed meshes.

Figure 2 shows a small collection of input images along with the intermediate and final output of a state-of-the-art SfM+MVS pipeline [SZPF16]. Note that the final output shows some clearly visible artifacts: (a) occluding objects (trees, cars...) have been included in the reconstructed mesh, (b) parts of the reconstructed mesh are missing, (c) parts of the surface have a curved shape, and (d) there is low detail in the mesh color.

Missing depth values mostly occur in facade parts lacking image features and on specular surfaces (e.g. windows) where consistency checks among images will fail. Since these features depend on surface reflectivity, adding more images from different views does not alleviate this problem.

Rounded shapes around the facade can be attributed to the surface reconstruction algorithm employed (e.g. Poisson-based approaches). These algorithms define an implicit function and then obtain the reconstructed surface by extracting an appropriate isosurface. Since the implicit function is assumed to be smooth, when the sampling rate is not enough to capture the facade details, the reconstructed surface shows a curved appearance.

Low-resolution color in the example shown in Fig. 2 is due to the use of per-vertex color. The use of texture maps increases the color resolution, but results are still not satisfactory due to shape offsets in the reconstructed surface that cause color from images to be mapped onto wrong surface locations.

In this paper we slightly modify the typical SfM+MVS pipeline to address the issues mentioned above. The new pipeline is depicted in Fig. 3. In particular, (1) we segment the (undistorted) input images into a small set of classes (constructions, sky, obstacles...), using a pre-trained CNN, (2) we remove from the depth maps those parts containing unintended objects, (3) we repair the depth map by detecting missing parts and assigning plausible depth values to them, and (4) we use an optional meshing algorithm that avoids

DOI: 10.2312/ceig.20181162

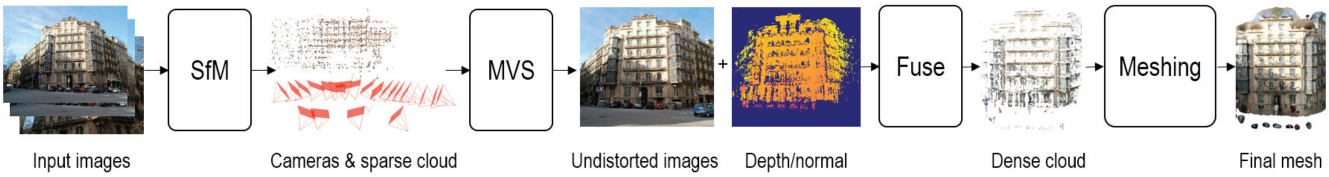

Figure 1: *Standard pipeline for 3D reconstruction from images.*

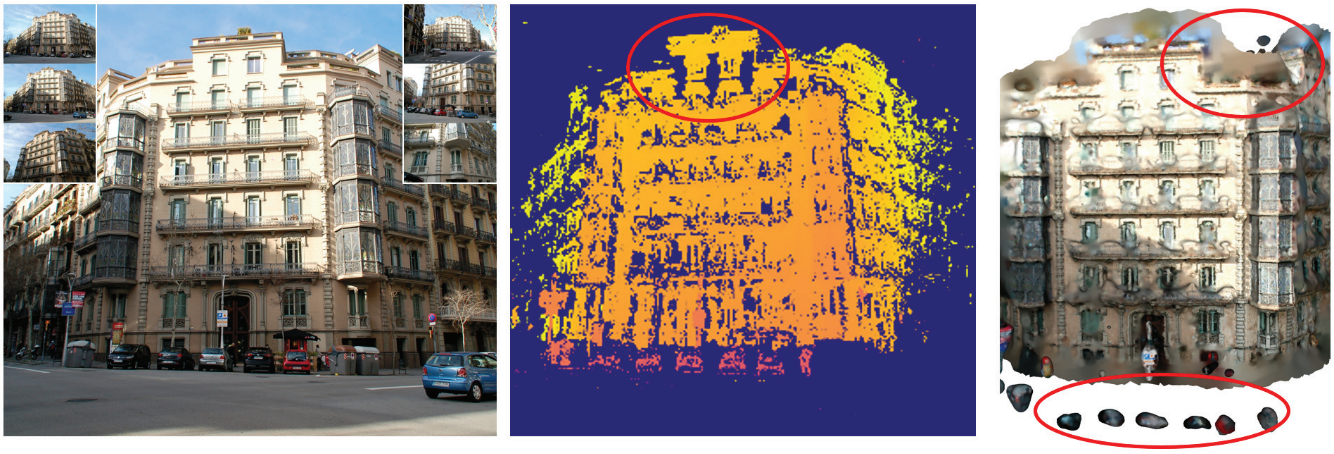

Figure 2: *Artifacts in standard SfM+MVS pipelines: input images, depth map and final mesh.*

surrounding curved shapes produced by Poisson-based approaches. The new pipeline is able to create detailed, plausible reconstructions of building facades, with much better shape and appearance than conventional SfM+MVS pipelines.

2. Previous Work

There is extensive research done in the reconstruction of 3D models from point clouds [BTS*14]. However, most of the developed techniques are designed for models obtained from laser scanning. As a result, they rarely use the information used to obtain the input point cloud, even when the method is integrated into a Multi-View Stereo pipeline.

It would also be possible to apply a repair technique after the reconstruction [Ju09], but disregarding all the information available during model capture, it is difficult to recover from certain errors. In particular, a commonly used prior in both reconstruction and repair algorithms is model smoothness, which for certain inputs can be counterproductive. In the case of the reconstruction of buildings, one possibility is to take advantage of their symmetries, as Ceylan et al [CMZP14] do. Still the running time of this type of methods can become prohibitive for massive models or when the number of models to process is large enough.

A common strategy in Multi-View Stereo algorithms is to make a local estimation of the depth maps of the input images using subsets of these. Afterwards the computed depth maps are integrated into a single surface rejecting all points that despite having a match show inconsistencies. Most of the work that has been done regarding the repair of these depth maps has focused on those situations in which the number of images is small. This is because the low redundancy of this type of datasets results in a reduction in the accuracy of the reconstruction. The quality of the meshes produced increases considerably if the number of outliers is reduced in the depths maps [CVHC08]. In any case, when the captured surface contains enough specular or translucent surfaces, as well as a good number of difficult-to-avoid occluders, it is necessary to repair the depth maps. When the number of images is high, it is also interesting that the repair algorithm is as efficient as possible.

The reconstruction process discards those points that do not pass the consistency tests. As a result, all surfaces with no texture features or that due to specular reflection show different colors depending on the viewpoint are not reconstructed correctly. As urban models can be approximated by a set of flat surfaces, it is possible to add planar priors as terms of consistency to the dense reconstruction process of the MVS pipeline. Mičušík and Košecká [MK10] introduced a method that used such planar priors to a set of superpixels computed on each input image. This same superpixel representation may be used to output a repaired mesh from any of the computed depth maps. An algorithm proposed by Bódis-Szomorú et al [BSRVG15] extracts a 2D mesh from a depth map, and reprojects its 2D points to 3D using the sparse point cloud that results from the SfM (Structure-from-Motion) step.

Another possibility is to detect any symmetry present in out input models and exploit it to repair them. Pauly et al [PMW*08] introduced the idea that repetitions of a part resulted in a grid of

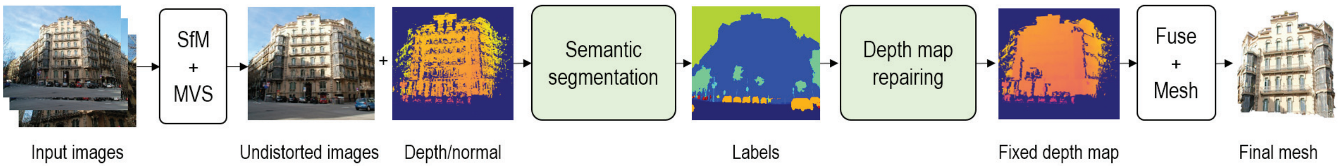

Figure 3: *Proposed pipeline for 3D reconstruction of buildings.*

points in the appropriate transformation space. This makes it possible to detect this regularity and apply it to repair all instances of a common surface part. Zheng et al [ZSW*10] adapted this process to point clouds obtained from terrestrial Lidar. On the other hand Li et al [LZS*11] improved this type of algorithms using the associated color information where available. Still some features typical of urban models cannot be dealt by this techniques alone and their performance make them prohibitive for large datasets.

In parallel, the introduction of depth capture devices like the Kinect has produced the need to process the resulting depth maps. In addition to filtering the noise in these datasets, it is essential to fill in the holes that are present [LJW14]. Nevertheless such techniques rarely include any prior knowledge specific to building reconstruction due to the low range of such devices.

3. Our approach

3.1. Image Segmentation

We use a state-of-the-art encoder-decoder network architecture (Xception-65 from DeepLab-v3+ [CPSA17]) to segment the undistorted images into the following classes: construction (building, wall and fences), flat (road, sidewalks), sky, obstacles (persons, vehicles, and objects such as poles and traffic lights), nature (vegetation, terrain) and sky. For this purpose, the network was trained with the well known ImageNet and Cityscapes datasets [COR*16]. Cityscapes is a large-scale dataset for semantic urban scene understanding that contains 25,000 annotated images from 20 different European cities, into a superset of the classes we use. The segmentation is applied once input images have been undistorted, so that there is a one-to-one mapping between depth map pixels and segmented image pixels. This will allow us to remove unwanted objects either directly on the depth maps, or later on in the dense point cloud. Fig. 4 shows one example of our automatic segmentation. Cars, trees, sky and building facades appear clearly segmented.

3.2. Repairing the depth maps

MVS techniques create a dense reconstruction of the scene through the computation of depth and normal maps for each input image. Unfortunately, depth maps often exhibit two major types of artifacts: unwanted obstacles such as trees and vehicles, and missing parts on uniform/mirror surfaces. The advantages of operating directly on the depth maps is two-fold: we can remove distracting objects that will not appear in the final dense pointcloud/meshes, and we can repair holes so that each individual depth map can be rendered with a high-quality texture map.

We remove unintended objects by invalidating (setting to 0) their values in the depth maps. By default, we remove pixels belonging to all classes except constructions (i.e. we remove roads, sidewalks, sky, obstacles, vegetation, and sky), although we let the user to preserve a subset of these classes (e.g. we might choose the preserve vegetation for vertical gardens).

Repairing missing parts in the depth proceeds as follows. We assume that missing values are set to 0. We first threshold the depth mask to get a binary mask B by distinguishing zero-valued pixels from the rest. Then we apply a border-following algorithm [S*85] to B to extract all the contours separating valid regions from non-data ones. Each contour c is represented as a collection of 2D point coordinates. Contours might include other contours and define deep inclusion hierarchies, with complex topologies (Fig. 5). Points inside a contour might have only valid depth values, only zero values, or a combination of both, due to non-valid regions surrounding valid ones and vice-versa. We consider a contour c to define a repairable region if (a) it contains at least one pixel with zero depth in its interior, (b) it contains pixels from the intended classes in the segmented image and (c) the area of the bounded region is below some threshold. The interior of repairable contours are fixed by replacing zero-depth values with a regression plane found through RANSAC on the 3D coordinates of the points surrounding the contour's boundary. In particular, we apply RANSAC to the 4-neighbors of all points of the contour boundary, excluding those with non-valid (zero) depth value. Since very small contours do not allow for a robust computation of the regression plane, we handle these separately by just using the average depth of valid neighboring points. Fig. 5 illustrates this process. Notice that most holes have been repaired, except those at the bottom-left of the image, where tree branches have been detected and the holes have not been repaired. Although this method can be applied to depth maps separately, we can also preprocess the depth maps to replace non-valid values using valid values from the rest of the registered depth maps (since camera poses are known), and then fill only the remaining holes using the regression plane approach described above.

3.3. Alternative meshing

Dense meshes from MVS are often created through surface reconstruction methods adopting an implicit function approach. These methods are robust against data noise, but produce water-tight surfaces even when the point cloud only captures a few sides of the object. In the case of building facades, state-of-the-art Poisson-based reconstructions tend to produce smooth surfaces around the facade (Fig. 2). Furthermore, color reproduction in these parts is often poor.

In some applications (e.g. restoration planning, cultural heritage

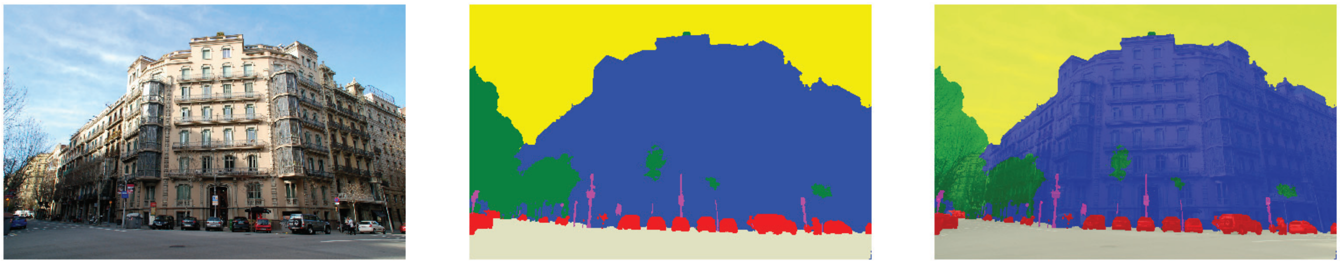

Figure 4: *Undistorted image and its semantic segmentation.*

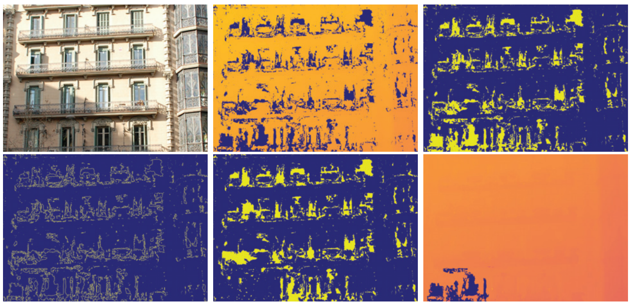

Figure 5: *Repairing holes in the depth map: image, depth map (using a color map for illustration), segmentation into valid (blue) and non-valid (yellow) values, extracted contours, mask defining repairable holes, and fixed depth map.*

documentation), it might be more desirable to get a 3D textured model directly from one of the reconstructed depth maps. A clear advantage is that undistorted images are readily available to be used as high-quality texture maps. We apply this technique to our repaired depth-maps, by simply applying displacement mapping to a tessellated plane, and directly using normalized image coordinates as texture coordinates to transfer color from the undistorted color images. We automatically remove faces at step discontinuities (as in [CL96]) and zero-valued parts of the facade.

4. Results

We tested the proposed pipeline on a PC equipped with an Intel i7-4770K CPU and an NVIDIA GTX 770 GPU. We used COLMAP for SfM (on CPU) and MVS (on GPU), TensorFlow (on GPU) for image segmentation, and Python (on CPU) for depth map repairing.

Original images (facades from Barcelona, Madrid and Zaragoza) had a resolution of 3008×2000 pixels. Depth maps were generated at 2000×1270 resolution due to the GPU memory constraints of MVS. Each image set contained between 6 and 12 images. The average running time for the standard COLMAP pipeline was about 12’ (4’ for SfM, 8’ for MVS). Adding our additional steps only added about 3’ to the process.

Figure 6 shows some examples of depth maps that have been automatically segmented through the Xception-65 encoder-decoder network trained on the Cityscapes dataset. Note how the building facade is clearly segmented from sky, trees and obstacles. Tree branches were detected despite branches having almost no leaves.

Fig. 7 compares the output of a state-of-the-art SfM+MVS pipeline (COLMAP), with that of our meshing algorithm. Notice that COLMAP creates a triangle mesh with per-vertex colors, whereas we use the undistorted images to texture the mesh. Fig 8 compares renders of a displacement-mapped surface before and after depth map repairing. In both cases we use the undistorted color image for texture mapping. Since missing data in the repaired depth map has been recovered, a single depth map already provides an acceptable representation of the facade. Fig 9 shows further examples of depth maps automatically repaired with our algorithm. Fig 10 shows our results after fusion [SZPF16] of the repaired depth maps. Windows and texture-less patches have been reconstructed successfully.

5. Conclusions

We have presented a simple extension of a common 3D reconstruction pipeline to generate high-quality renders of reconstructed facades. The main idea is to use SfM+MVS to recover the depth maps of the input images, and to repair them by removing unwanted objects (through semantic segmentation) and filling holes through local regression planes. Repaired depth maps can be used to generate clean point clouds or, alternatively, to render high-quality meshes through a combined displacement mapping + texture mapping technique. As future work, we plan to further explore the fusion and meshing steps with repaired depth maps.

Acknowledgments

This work has been partially funded by the Spanish Ministry of Economy and Competitiveness and FEDER under grant TIN2017-88515-C2-1-R, and the Spanish Ministry of Education, Culture and Sports PhD grants FPU13/01079 and FPU14/00725.

References

[BSRVG15] Bódis-Szomorú A., Riemenschneider H., Van Gool L.: Superpixel meshes for fast edge-preserving surface reconstruction. In *Proceedings CVPR 2015* (2015), pp. 2011–2020. 2

[BTS*14] Berger M., Tagliasacchi A., Seversky L., Alliez P., Levine J., Sharf A., Silva C.: State of the art in surface reconstruction from point clouds. In *EUROGRAPHICS star reports* (2014), pp. 161–185. 2

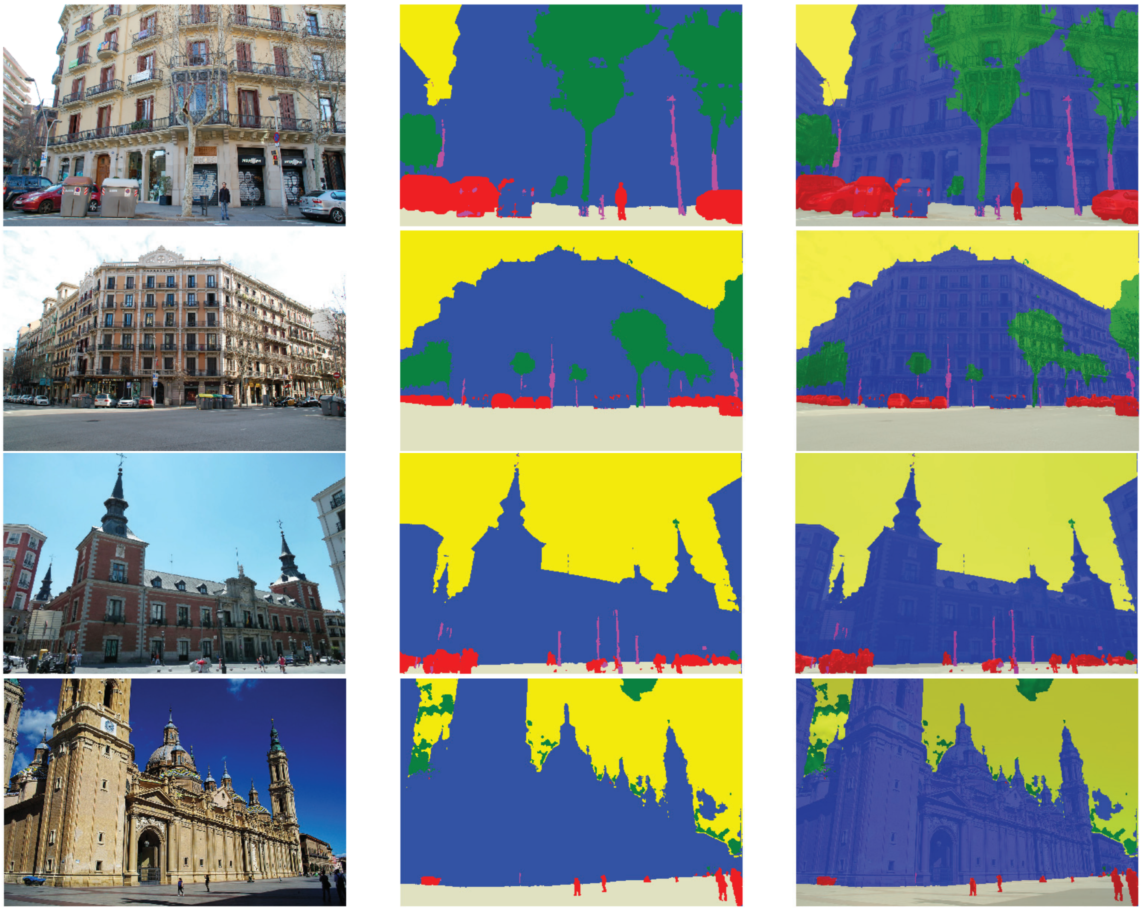

Figure 6: *Images segmented automatically prior to depth cleaning.*

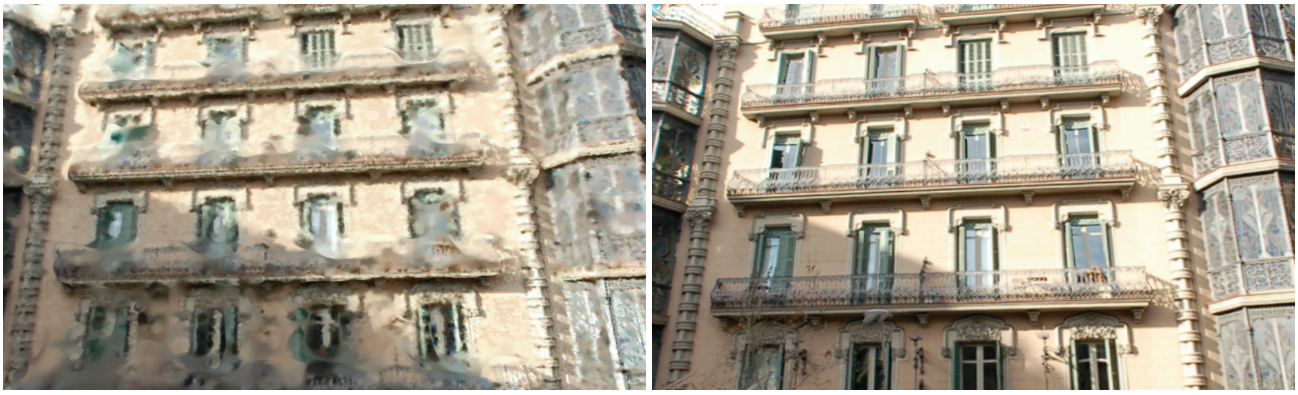

Figure 7: *Renders of the meshes produced by COLMAP (left) and our meshing algorithm (right).*

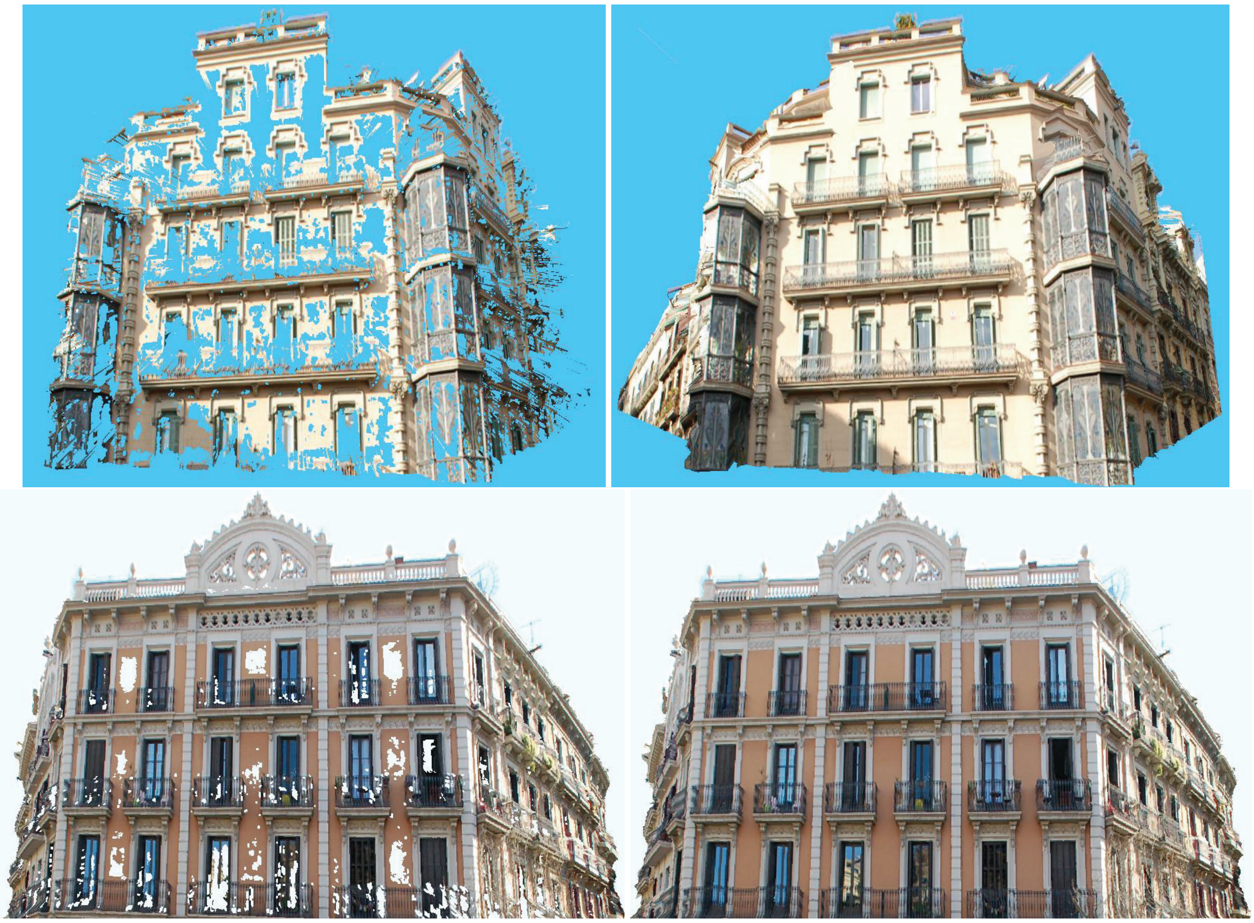

Figure 8: *Meshes before (left) and after (right) applying our depth repairing algorithm.*

[CL96] Curless B., Levoy M.: A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 303–312. 4

[CMZP14] Ceylan D., Mitra N. J., Zheng Y., Pauly M.: Coupled structure-from-motion and 3d symmetry detection for urban facades. *ACM Trans. Graph. 33*, 1 (2014), 2:1–2:15. 2

[COR*16] Cordts M., Omran M., Ramos S., Rehfeld T., Enzweiler M., Benenson R., Franke U., Roth S., Schiele B.: The cityscapes dataset for semantic urban scene understanding. *CoRR abs/1604.01685* (2016). arXiv:1604.01685. 3

[CPSA17] Chen L.-C., Papandreou G., Schroff F., Adam H.: Rethinking atrous convolution for semantic image segmentation, 2017. arXiv:1706.05587. 3

[CVHC08] Campbell N. D. F., Vogiatzis G., Hernández C., Cipolla R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision – ECCV 2008* (2008), Springer Berlin Heidelberg, pp. 766–779. 2

[Ju09] Ju T.: Fixing geometric errors on polygonal models: A survey. *Journal of Computer Science and Technology 24*, 1 (Jan 2009), 19–29. 2

[KH13] Kazhdan M., Hoppe H.: Screened poisson surface reconstruction. *ACM Transactions on Graphics (TOG) 32*, 3 (2013), 29. 1

[LJW14] Le A. V., Jung S.-W., Won C. S.: Directional joint bilateral filter for depth images. In *Sensors* (2014). 3

[LZS*11] Li Y., Zheng Q., Sharf A., Cohen-Or D., Chen B., Mitra N. J.: 2d-3d fusion for layer decomposition of urban facades. In *Computer Vision (ICCV), 2011 IEEE International Conference on* (2011), IEEE, pp. 882–889. 3

[MK10] Mičušík B., Košecká J.: Multi-view superpixel stereo in urban environments. *International journal of computer vision 89*, 1 (2010), 106–119. 2

[PMW*08] Pauly M., Mitra N. J., Wallner J., Pottmann H., Guibas L. J.: Discovering structural regularity in 3d geometry. *ACM transactions on graphics (TOG) 27*, 3 (2008), 43. 2

[S*85] Suzuki S., et al.: Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing 30(1)* (1985), 32–46. 3

[SF16] Schönberger J. L., Frahm J.-M.: Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 1

[SZPF16] Schönberger J. L., Zheng E., Pollefeys M., Frahm J.-M.: Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)* (2016). 1, 4

[ZSW*10] Zheng Q., Sharf A., Wan G., Li Y., Mitra N. J., Cohen-Or D., Chen B.: Non-local scan consolidation for 3d urban scenes. In *ACM SIGGRAPH 2010 Papers* (2010), SIGGRAPH '10, ACM, pp. 94:1–94:9. 3

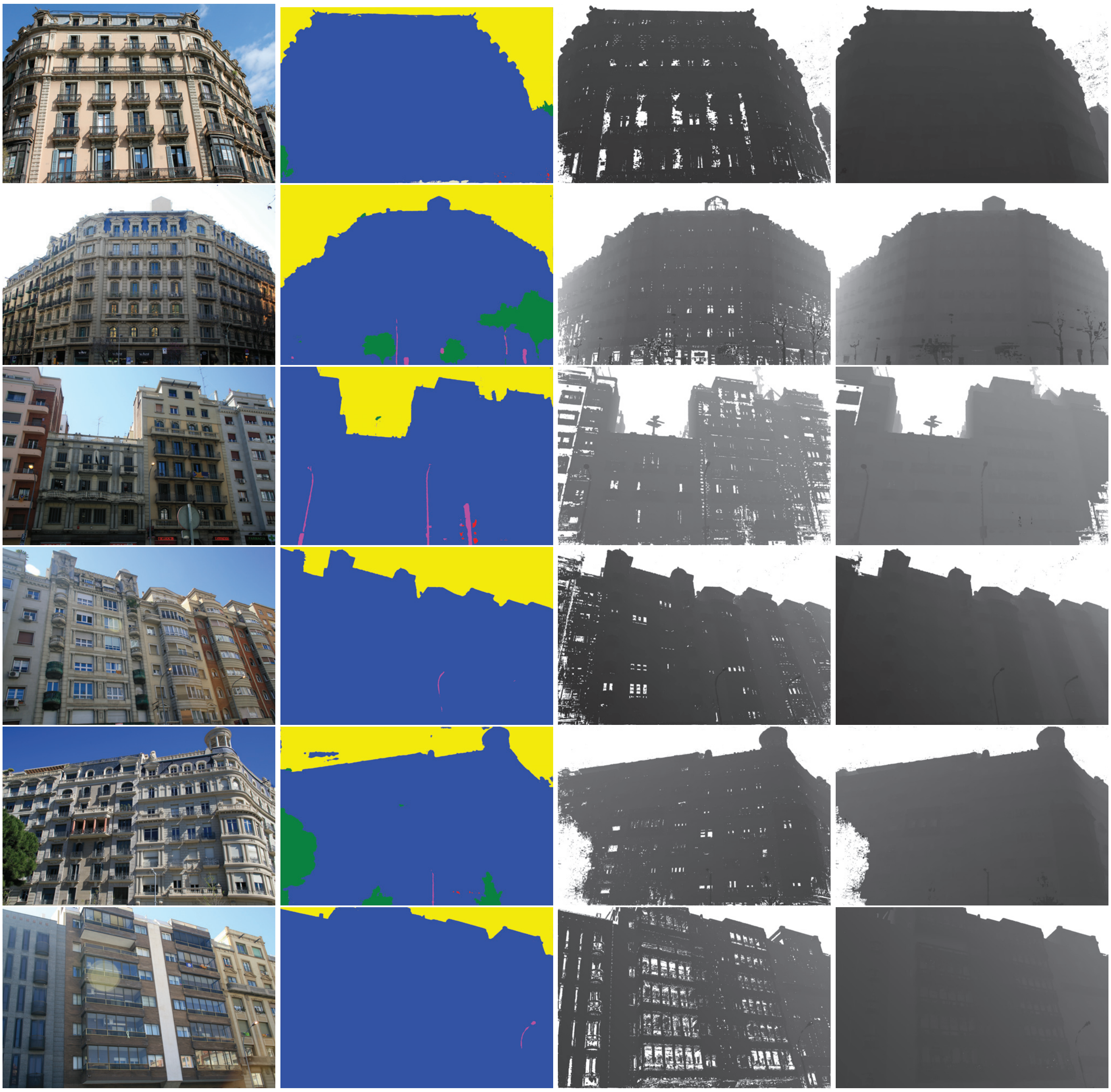

Figure 9: *Repairing results: RGB image, segmentation, original depth map, repaired depth map.*

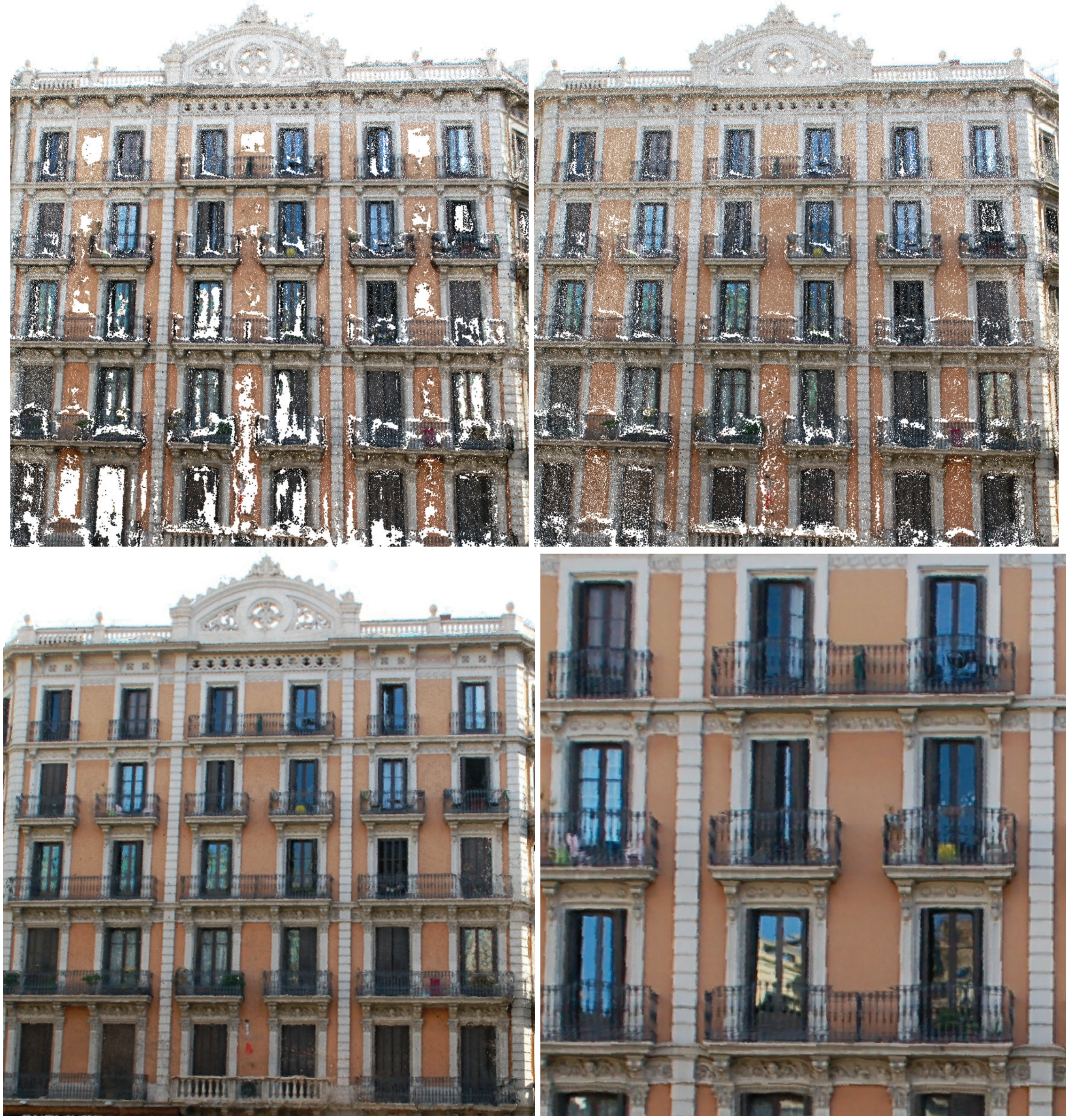

Figure 10: *Impact of depth map repairing: point cloud from original depth maps, point cloud from repaired depth maps, reconstructed mesh, and close-up view with projective texture mapping.*