

# CHALLENGES IN FUSION OF HETEROGENEOUS POINT CLOUDS

Fabio Bracci, Martin Drauschke\*, Stefan Kühne, Zoltán-Csaba Márton

German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Wessling, Germany  
{Fabio.Bracci, Martin.Drauschke, Stefan.Kuehne, Zoltan.Marton}@dlr.de

Commission II, WG II/3

**KEY WORDS:** Point Cloud Fusion, 3d Reconstruction, Registration, ICP, Parallel Planes

## ABSTRACT:

Different platforms and sensors are used to derive 3d models of urban scenes. 3d reconstruction from satellite and aerial images are used to derive sparse models mainly showing ground and roof surfaces of entire cities. In contrast to such sparse models, 3d reconstructions from UAV or ground images are much denser and show building facades and street furniture as traffic signs and garbage bins. Furthermore, point clouds may also get acquired with LiDAR sensors. Point clouds do not only differ in the viewpoints, but also in their scales and point densities. Consequently, the fusion of such heterogeneous point clouds is highly challenging. Regarding urban scenes, another challenge is the occurrence of only a few parallel planes where it is difficult to find the correct rotation parameters. We discuss the limitations of the general fusion methodology based on an initial alignment step followed by a local coregistration using ICP and present strategies to overcome them.

## 1. INTRODUCTION

Urban environments can efficiently get reconstructed in 3d from images by photogrammetric computer vision technologies such as research prototypes (Furukawa and Ponce, 2010, Irschara et al., 2012, Kuhn et al., 2017) or commercial products, e.g., Pix4D, nFrames, Agisoft or Acute3D. Depending on the image source, the reconstructed point clouds show different building parts and have different point densities and accuracies. For instance, (Rupnik et al., 2018) derive digital surface models from *very high resolution* satellite images with a ground sampling distance of 0.3 m. I.e., their landscape model may have a point density of 9 points per m<sup>2</sup> only. In contrast, we derive point clouds from terrestrial images or low-flying UAVs with several thousand 3d points per m<sup>2</sup>. Alternatively to reconstructed point clouds, they also can be acquired by (mobile) laser scanning systems as used in (Munoz et al., 2009), (Lauterbach et al., 2015) or (Vosselman et al., 2017).

Fusing such heterogeneous point clouds is a very active and very challenging research topic. Motivation can be to improve 3d models by combining different sources (ground and aerial perspective) or to insert a dense building model as point of interest into a sparser landscape model. This problem can be solved in three different ways. First, if images are available, the joint orientation of images from different sensors can be employed. (Koch et al., 2016) and (Roth et al., 2017) propose strategies to combine images with large scale differences and large perspective changes, respectively. Second, 2d features in the images and 3d features in the point clouds can be analyzed to find correspondences for fusing different models, cf. (Schenk and Csathó, 2002) or (Mishra and Zhang, 2012). Third, the point clouds are coregistered based on 3d information of the points themselves. A review on recent approaches in this field is given in (Pomerleau et al., 2015).

To be as general as possible with respect to the nature of point clouds, we are interested in the third way: fusing on the basis of point cloud coregistration. Thereto, a standard fusion methodology exists, and it consists of three steps: first initial alignment,

second local coregistration, and third surface smoothing and detection of artifacts. We discuss this workflow regarding challenging heterogeneous point clouds of urban scenes. Hence, the following technical problems have to be solved:

1. Point clouds can be very large. We obtain point clouds from reconstruction with more than 10 million points. Such amounts of data bring difficulties with manipulation in a viewer. This open issue remains untouched by us. Furthermore, we remark that all calculations are time and memory intensive, if the input datasets are very large.
2. Point clouds potentially have very large differences in extension and scale. On the one hand we have reconstructed point clouds from a satellite scene which may cover an entire city, i.e., their range is in kilometers. On the other hand we have close range reconstructions of scenes of few meters only. Scale differences may occur due to reconstruction output in absolute and relative coordinates. We overcome these difficulties by cropping point clouds to the area of interest and by manually performing an initial alignment step.
3. Urban scenes often show many parallel planes. Small translations and rotations do not change significantly the overlap region, which makes the search for an optimal fusion difficult. We address this problem in sec. 3.1.
4. Point clouds vary strongly in their point densities and their accuracies. This is due to, e.g., the sensor size or the distance between sensor and object, which are usually larger for acquisitions with airplanes or satellites. We address this problem in sec. 3.2.
5. Point clouds have only small overlaps. Point clouds from aerial data acquisitions only show ground and roof surfaces, while point clouds derived from campaigns on the ground show only building facades and street furniture. Hence, such point clouds only overlap around the buildings' footprints and eaves.

\*Corresponding author.

This paper is structured as follows. In the next section, we introduce and discuss the standard fusion methodology. Furthermore, we present strategies to overcome its limitations for challenging datasets. In sec. 3, we demonstrate the success of our proposed strategies in experiments with difficult datasets. Last, we conclude with a short summary and give an outlook on future work.

## 2. STANDARD FUSION METHODOLOGY

Generally, the fusion of two point clouds consists of three steps. First, both point clouds must be initially aligned to get a rough approximation for a common coordinate system and scale. Second, the local coregistration is performed. In the third step, surfaces are smoothed and outliers are reduced, and artifacts are removed or highlighted. We realized this general fusion strategy, and we discuss each step in this section. In our implementation, the initial alignment has to be performed manually, because we need a good approximation for the coregistration to avoid local minima in the following step of local coregistration. All other steps are fully automated.

We demonstrate this methodology on an example of two point clouds derived from images acquired by a UAV, cf. fig. 1. One dataset only contains nadir views, the other one only oblique views. Consequently, both point clouds overlap in the ground and roof areas, but the first point cloud does not show many points of the facades. Our 3d reconstruction pipeline consists of a highly accurate image orientation (Michelini and Mayer, 2016), dense image matching using semi-global matching – SGM (Hirschmüller, 2008) and the scalable reconstruction approach of (Kuhn et al., 2017).

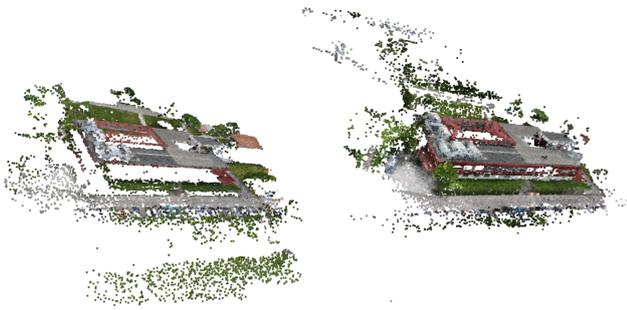


Figure 1. Two point clouds of the same building. The point clouds with RGB texture are derived by 3d reconstruction from nadir views (left) and from oblique views (right).

### 2.1 Initial Alignment

If both point clouds are not given in a common global coordinate system, e.g., GPS, one point cloud must be transformed into the coordinate system of the other. Standard implementations such as the Point Cloud Library (Aldoma et al., 2012) only succeed, if the point clouds have a significant overlap and the same scale. If no assumptions can be done, this initial alignment step must be performed under manual supervision. Selecting at least three homologous, not collinear points in both point clouds, cf. fig. 2, we are able to determine the rotation, scale and translation parameters by determining a Helmert transform. This way, we are able to estimate the transformation of the coordinate system using a least squares optimization when selecting more than three points.

### 2.2 Local Registration

Local registration is often used for refining the overlap between two point clouds. Then, a distance between both data sets is

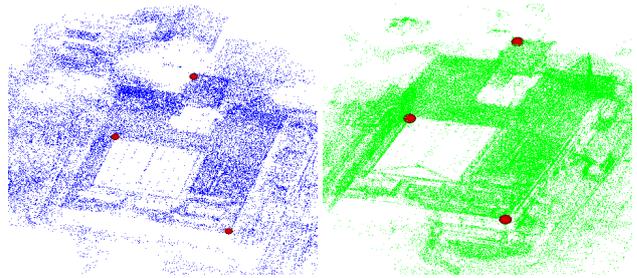


Figure 2. Two point clouds from fig. 1 with three homologous points. The markers were manually selected without identifying individual points of the clouds.

minimized, often iteratively in a EM-based framework, because a closed form solution is not known. One standard approach is the Iterative Closest Point – ICP (Besl and McKay, 1992), where correspondences between point pairs are identified and the distance between all corresponding point pairs is minimized. There exist many implementations of ICP, e.g., (Li and Hartley, 2007) or (Rusu and Cousins, 2011). Such implementations vary in the identification and minimization steps which have to get adjusted to the certain application. Usually, the points are analyzed within a neighborhood to identify corresponding point pairs. Then those pairs are filtered using different heuristics, and often a least squares optimization is performed to minimize the, e.g., point to point or point to plane distances.

Normal Distributions Transform (NDT) as proposed by (Biber and Strasser, 2003) and improved by (Magnusson et al., 2007) is an alternative to ICP. In NDT, the spatial densities of the datasets are modeled by Gaussian mixture models, and the overlap between both mixture models is maximized. In our experiments, NDT was much faster than ICP, but it was less robust, requiring a more accurate initial alignment. Thus, we only document the experiments with ICP in this paper. Fig. 3 shows the two point clouds after the initial alignment step (top) and after employing ICP (bottom) based on its standard implementation in the Point Cloud Library (Rusu and Cousins, 2011).

Nevertheless, ICP has limited success for some challenging datasets. If both point clouds have significantly different point densities, then the algorithm is unable to find correct point-to-point correspondences. Furthermore, the search for the optimal fit will get stuck in a local minimum, if there are no further constraints. This problem is essential for symmetric scenes and for scenes with many parallel planes.

In case of symmetries or multiple parallel planes, small rotations and translations do not affect the quality of overlap for most of the points. Thus, the optimization converges to a local minimum of a flat valley in the cost function. Such a case occurs for the scene in fig. 3, where the ground and the parallel rooftop levels constitute most of the correspondences. Therefore in-plane translations and rotations do not affect the overall optimization metric enough to produce a gradient to the optimal alignment. We can solve this problem by extracting planar surfaces (Rabbani et al., 2006) and their boundaries to perform ICP only on boundary points. I.e., we will not consider the large amount of uninformative points for the in-plane rotations and translations. Results for this degenerate case are presented in the sec. 3.1.

### 2.3 Surface Smoothing & Artifacts

In this subsection we describe all aspects of point cloud smoothing. The goal is to obtain surfaces with less noise and merge the two clouds. Remark that we do not determine real meshes, e.g.,

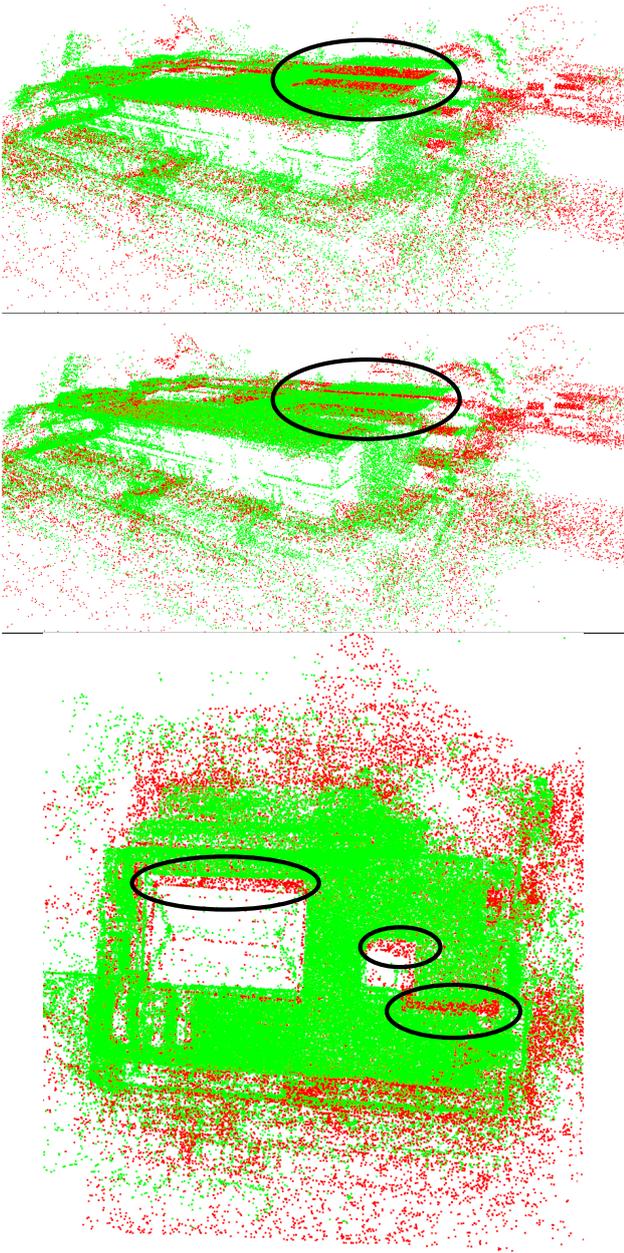


Figure 3. Result of the initial alignment (top) and the local registration with ICP (middle and bottom). Comparing top and middle view, the offset is significantly smaller after employing ICP, cf. the region of a roof super structure marked by the ellipses. Additionally, we highlighted regions in the bottom view which show the limitations of ICP, because its result is not optimal.

as done by (Liu et al., 2007), (Bodenmüller, 2009) or (Caraffa et al., 2016), and we also do not segment the point cloud, e.g., as done by (Nguattem and Mayer, 2016) or reviewed by (Grilli et al., 2017), but we locally estimate the object surfaces by polynomials.

We will note the source point cloud  $Q$  and the target point cloud  $P$ . If both point clouds have similar properties such as accuracy or point density, it does not matter which one is source and which one is target. Otherwise, we advice to consider the point cloud with lower quality to become the source point cloud. We approximate the surfaces through the points in  $P$  by piecewise polynomial functions and project the points of  $Q$  onto these, i.e., we project low-quality points onto the high-quality target surface

following (Márton et al., 2009).

For each point  $q \in Q$  a local neighborhood  $P_q$  is selected, by using either a fixed number of nearest neighbors, or all neighbors in a fixed radius. The first method can result in neighbors that are relatively far away from the currently fitted point, and their influence must be minimized when fitting the point (e.g. through the distance-based weighting, see below). The second method has the disadvantage that it may yield too few neighbors if the search radius is too small, but it ensures that the neighbors will surround the point to be fitted. In our approach the second method is preferred, as we require a maximum distance as cutoff to avoid changing points that have no correspondence in case  $Q$  and  $P$  are different (i.e. in non-overlapping areas). Whereas, using the first method requires extra filters to ensure this.

Weights are assigned to every neighbor  $p_i \in P_q$  of  $q$  based on the distance to the currently fitted point (with a scaling parameter  $s$ , that can be chosen depending on the search radius):

$$w_i = e^{-\|p_i - q\|/s}, \quad q \in Q, \quad p_i \in P_q \quad (1)$$

Additionally, points can be weighted based on their estimated accuracy as well.

Each point  $q \in Q$  will be projected onto the plane  $k_q$  fitted through its nearest neighbors  $P_q$ , thus bringing it in the proximity of the surface. This plane  $k_q$  defines a local reference frame, having the projection of the query point as the origin and with axes denoted  $u, v, n$  (where  $u$  and  $v$  are coordinates in the local coordinate system lying within the plane, and  $n$  is the normal of the plane).

In order to fit a weighted least-squares plane, we assemble a weighted covariance matrix from the  $m$  points  $p_i \in P_q$ :

$$C = \sum_{i=1}^m w_i \cdot (p_i - \bar{p}) \cdot (p_i - \bar{p})^T, \quad \bar{p} = \frac{1}{m} \cdot \sum_{i=1}^m p_i \quad (2)$$

followed by the eigenvalue and eigenvector computation.

In the next fitting step the polynomial approximation is performed. For this, the standard coordinate system is transformed into the local coordinate system  $u, v, n$  and a high-order bi-variable polynomial is fitted to the heights of the points above the plane and the height of the projected point is recalculated. A polynomial of order  $N$  has  $(N + 1) \cdot (N + 2)/2$  terms and is of the form:

$$h_q(u, v) = \sum_{a=0}^N \sum_{b=0}^a c_{a(a+1)/2+b} \cdot u^{a-b} \cdot v^b. \quad (3)$$

To compute the vector of unknown coefficients  $c_i$  of the polynomial, we minimize a weighted least squares error function as in (Márton et al., 2009), using a recomputed  $w$  as weights based on the initial projection of  $q$ , i.e., the origin of the new frame of reference.

After resampling, surface normals have to be re-computed as the normal of the fitted polynomial in point  $q$ , through the partial derivatives. We can easily compute the normal  $n$  of the estimated surface by computing the two partial derivatives at  $(u, v) = (0, 0)$  in the local coordinate system and their cross product:

$$n = (u_0 + c_1 \cdot n_0) \times (v_0 + c_2 \cdot n_0). \quad (4)$$

where  $c_1$  and  $c_2$  are the first order coefficients for the  $u$  and  $v$  coordinates, respectively,  $u_0$ ,  $v_0$  and  $n_0$  being the corresponding unit vectors.

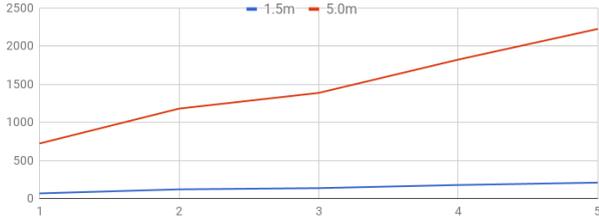


Figure 4. Time (in sec.) for calculating polynomial models in a neighborhood of 1.5 (blue) and 5 m (red) of orders 1-5 (on the horizontal axis) for the dataset in fig. 10.

Fig. 4 shows the effect of  $N$  and search radius on the runtime. A large radius enables the merging of distant misaligned surfaces, while the distance-based weighting ensures that the smoothing is not affected by distant points in case there is good overlap. A high polynomial order increases accuracy, but quickly becomes impractical for large radii. Since there need to be at least as many points in a neighborhood as terms in eq. 3, the lower limit of the search radius is influenced by  $N$ .

If we note  $\mathbf{q}'$  as projection of  $\mathbf{q}$  onto the polynomial surface  $h_{\mathbf{q}}$ , the Euclidean distance between  $\mathbf{q}$  and  $\mathbf{q}'$  is a good measure for the distortion introduced by the fusion. It can be used to detect (i) inconsistent measurements in the two point clouds, e.g., because one of them is inaccurate, (ii) fusion artifacts, e.g., when the alignment did not converge to a correct solution and produced duplicated surfaces, or (iii) the amount of deviation from the estimated surface (due to noise) that was smoothed out. In consequence, artifacts could be marked in the fused point cloud, if a threshold for the distance is exceeded.

### 3. EXPERIMENTS ON CHALLENGING DATASETS

In this section we describe our experiences on fusing challenging datasets. The first case shows a building scene with a high overlap, but the overlapping area in the complete scene consists only of a few parallel planes. The second case is about fusing a sparse landscape model derived from satellite images and a dense, but small LiDAR point cloud acquired by a mobile scanning system.

#### 3.1 Fusing Point Clouds Overlapping only at Parallel Planes

Regarding the dataset of two point clouds from a building (fig. 1), we observed a small offset after employing the standard ICP of the Point Cloud Library – PCL (Rusu and Cousins, 2011), cf. fig. 3. As discussed in sec. 2.2, this is because the overlapping region consists mostly of parallel planar surfaces. Instead of using all points for ICP, we determine all boundary points and perform ICP only on these points. We extract planar surfaces with a simple region growing method that considers the estimated surface normals, and the boundaries of these surfaces, both using implementations from PCL. The intermediate results are shown in fig. 5.

Then, performing another ICP optimization on the contours will not consider the large amount of uninformative points for the in-plane rotations and translations. Using the 2d version of the optimizer from PCL’s registration API ensures that only the remaining three degrees of freedoms are changed, but requires a transformation of the clouds such that to align the planes’ normal to the vertical coordinate axis. Therefore the regular ICP was used in this step as well, as out-of-plane correspondences are highly unlikely when using the contours.

The final result is shown in fig. 5 (bottom), where contour points of both point clouds are visualized in red. Both point clouds look perfectly coregistered.



Figure 5. Parallel planar surfaces constituting the majority of the overlapping area (top and middle). The correction based on the plane’s contour points (bottom) produces an improvement over the alignment from fig. 3, in spite of dataset-specific differences.

#### 3.2 Fusing Landscape and Building Model

The dense point cloud as shown in fig. 6 has been acquired by a mobile LiDAR system. It consists of more than 800 000 points showing the road and buildings facades of a downtown crossroads in Germany. The ground resolution at the road is less about 1 cm due to the short distance of the scanner to the ground.

The much sparser landscape model is derived from *WorldView-01* satellite images following (Hirschmüller, 2011). We manually cropped to scene to the same extend as shown in the dense point

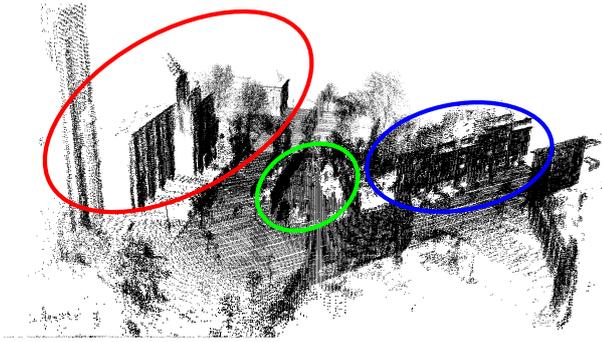


Figure 6. LiDAR point cloud of a downtown crossroads. We marked identical buildings to fig. 7 by ellipses of same color.

cloud, cf. fig. 7 (bottom). The cropped landscape model consists of almost 50 000 points and has a ground resolution of only 0.55 m per pixel. Originally, a digital surface model (2.5d) has been produced, which we transformed into a 3d point cloud. The reconstructed mesh shows only the roofs and the road surfaces, but it lacks of building facades. Furthermore, the accuracy of the height values is very low, e.g., the church tower is missing in the data.

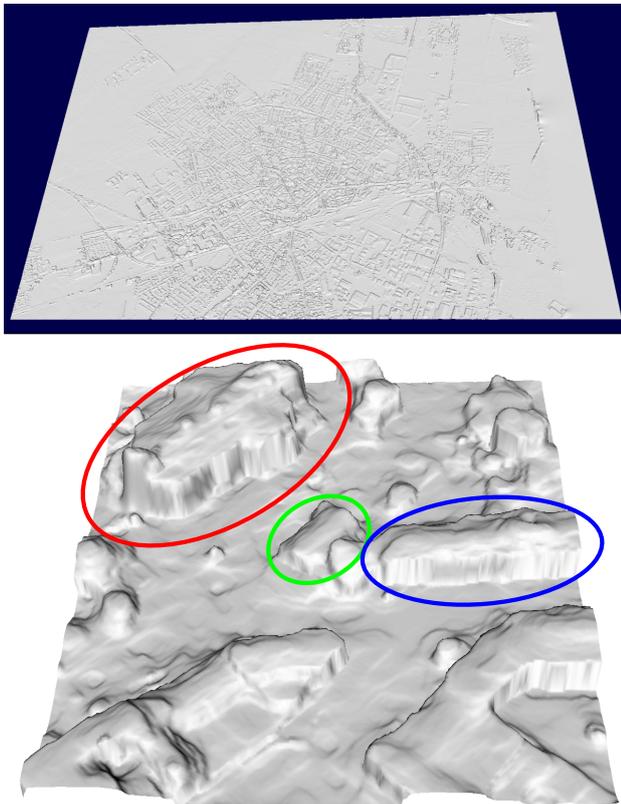


Figure 7. Complete (top) and cropped (bottom) surface model reconstructed from satellite images. We marked identical buildings to fig. 6 by ellipses of same color. The building surrounded by a red ellipse is a church whose tower has not been reconstructed from satellite images.

The result of the fusion of both datasets employing the standard ICP without surface smoothing is shown in fig. 8. At top, we show both point clouds in monochrome colors: green is the LiDAR point cloud, red is the sparse model derived from satellite images. The LiDAR point cloud of the building and the landscape model overlap almost exclusively at the road's surface. At bottom, we chose the same view from above but the colors de-

scribe the point heights (blue is high, red is low). Together both views allow a visual inspection of the fusion result. We also can recognize a drift in the LiDAR point cloud because of scanning errors, cf. the unaligned building facades marked by ellipses. The same result is presented in fig. 9, but we chose a side view on the scene. Here, we can inspect the match of height values on the street level of the result, which we regard satisfactory.

The height profile of the satellite scene varies highly, so that the ground looks corrugated. Furthermore roofs are often too low. Since both point clouds only overlap at the ground, the satellite model can be mapped on the LiDAR points along the ground. No correction can be done at other parts of the point clouds without making assumptions about the nature of the scene.

In the surface smoothing step, points can be projected on polynomial surfaces of the denser dataset after performing ICP. We calculated the polynomial functions based on the LiDAR points for optimal mapping values. A distance-based weighting of points is triggered by a Gaussian function to decrease the influence of points far away from the centroid. Therefore we can safely use neighborhoods which are ten times larger than the estimated noise, i.e., in our case 1 to 2 m.

While it is computationally more expensive to fit higher order polynomials, in most cases the results are better especially if one considers the correctness of the recalculated surface normals and edges and corners. Fitting higher order polynomials is more and more costly as the number of members increases (see 4) but the corners' accuracy is maintained well at order 5 while still smoothing the large undulations of the satellite measurements of the road's surface, cf. fig. 10.

#### 4. SUMMARY AND OUTLOOK ON FUTURE WORK

We discussed the standard methodology for fusing point clouds using ICP. It does not perform well, if we want to fuse point clouds which have significantly different quality, and if the scenes consist of many parallel planes. For both problems we presented a solution, and we demonstrated their success.

In the experiment with the city model derived from satellite images and the LiDAR point cloud, we showed that we also are able to handle scenes with little overlap. The overlap can be even less, if a point cloud reconstructed from facade images taken from the ground shall be fused with point cloud derived from nadir views from above. Then we have a minimal overlap of only sharing the building's footprint and its eaves.

Solving this problem could include use of a method based on signed distance functions as in volumetric mapping approaches (Wagner et al., 2014), where occlusion information can be considered during optimization. These, however, are limited in their volume or resolution, due to the explicit representation of the 3D space, but methods for a hierarchic representation using octrees and their compression exist already (Hornung et al., 2013).

To improve usability and transparency for the user, the degenerate case of overlap between datasets consisting only of parallel planar surfaces should be detected automatically. Similarly, the degree of smoothing necessary could be estimated from the residual error metric of ICP. Also it remains to be analyzed which type of artifact visualization is the most informative for non-expert users.

Furthermore, we would like to analyze recently published local registration methods, e.g., (Zhou et al., 2016) and (Förstner and Khoshelham, 2017), or other ICP variants or implementations,

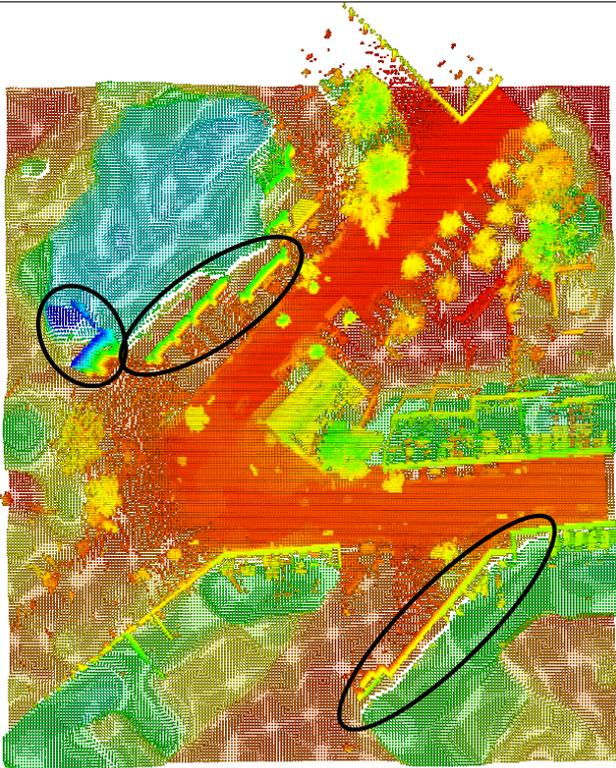
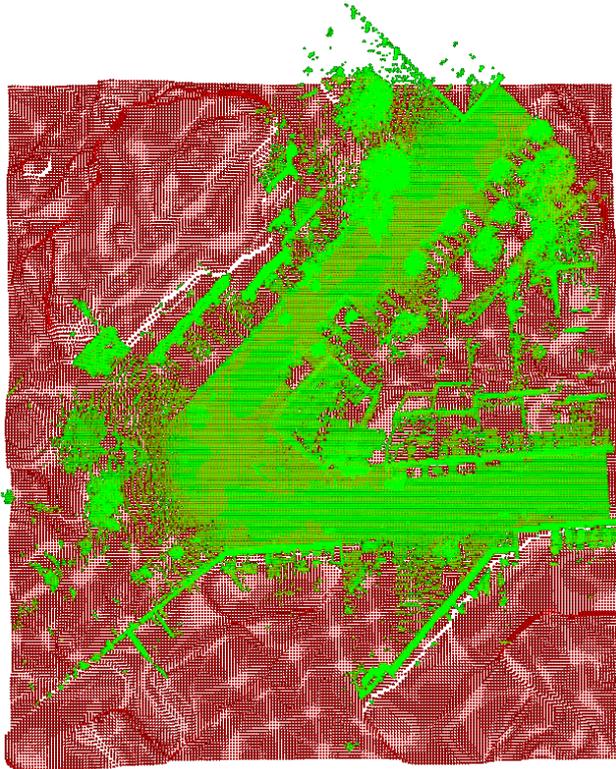


Figure 8. Result of fusion with standard ICP without surface smoothing. Top: Monochrome colors (green: LiDAR point cloud, red: landscape model derived from satellite images). Bottom: Same view with colors encoding point heights. Notice the height discrepancy at the boundary between facades, which are visible only in the LiDAR-based point cloud (appearing as dense lines), and the rooftops which are visible only in the satellite images (appearing as points arranged in a grid pattern).

e.g., (Geiger et al., 2012) and (Pomerleau et al., 2013). Last but not least, it would lead to better results of ICP, if semantic information as provided by (Weinmann et al., 2014), (Kuhn et al., 2016) or (Wang et al., 2017) would be considered for finding point-to-point correspondences, i.e., points classified as building or street would only be registered to points of the other dataset which have the same classification result.

## ACKNOWLEDGEMENT

We thank our colleague Iris Grixia for processing the satellite images to generate the digital elevation model. We also want to thank the reviewers for their helpful comments.

## REFERENCES

- Aldoma, A., Márton, Z.-C., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R. B., Gedikli, S. and Vincze, M., 2012. Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation. *IEEE Robotics & Automation Magazine* 19(3), pp. 80–91.
- Besl, P. J. and McKay, N. D., 1992. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), pp. 239–256.
- Biber, P. and Strasser, W., 2003. The Normal Distributions Transform: A New Approach to Laser Scan Matching. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. III: 2743–2748.
- Bodenmüller, T., 2009. Streaming Surface Reconstruction from Real Time 3D Measurements. PhD thesis, Technische Universität München.
- Caraffa, L., Brédif, M. and Vallet, B., 2016. 3D Watertight Mesh Generation with Uncertainties from Ubiquitous Data. In: *Computer Vision – ACCV 2016, Lecture Notes in Computer Science*, Vol. 10114, pp. 377–391.
- Förstner, W. and Khoshelham, K., 2017. Efficient and Accurate Registration of Point Clouds with Plane to Plane Correspondences. In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2165–2173.
- Furukawa, Y. and Ponce, J., 2010. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(8), pp. 1362–1376.
- Geiger, A., Lenz, P. and Urtasun, R., 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361.
- Grilli, E., Menna, F. and Remondino, F., 2017. A Review of Point Clouds Segmentation and Classification Algorithms. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLII-2/W3, pp. 339–344.
- Hirschmüller, H., 2008. Stereo Processing by Semi-Global Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(2), pp. 328–341.
- Hirschmüller, H., 2011. Semi-Global Matching: Motivation, Developments and Applications. In: *Photogrammetric Week 11*, pp. 173–184.

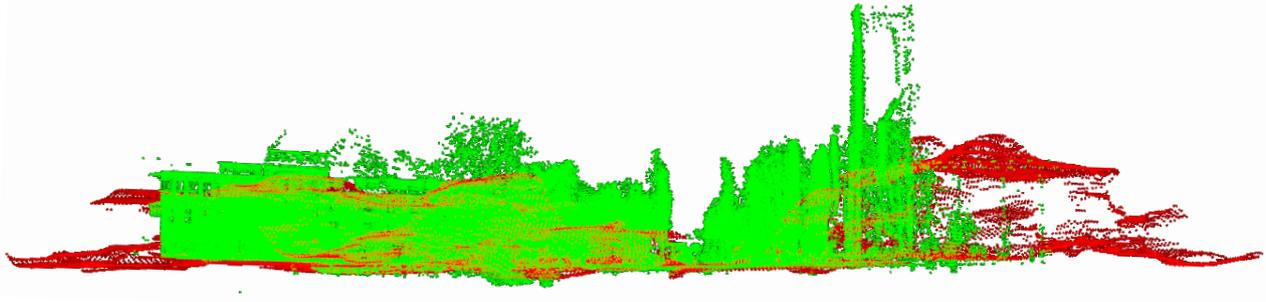


Figure 9. Result of fusion with standard ICP without surface smoothing. Side view presentation of same dataset as shown in fig. 8 (top) in a monochrome colors (green: LiDAR point cloud, red: landscape model derived from satellite images). One can notice the increased point density at the horizontal road's level, and also the large inaccuracy of the satellite height measurements at the rooftops.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C. and Burgard, W., 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots* 34(3), pp. 189–206. Software available at <http://octomap.github.com>.

Irschara, A., Rumpler, M., Meixner, P., Pock, T. and Bischof, H., 2012. Efficient and Globally Optimal Multi View Dense Matching for Aerial Images. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. I-3, pp. 227–232.

Koch, T., Zhuo, X., Reinartz, P. and Fraundorfer, F., 2016. A New Paradigm for Matching UAV- and Aerial Images. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. III-3, pp. 83–90.

Kuhn, A., Hirschmüller, H., Scharstein, D. and Mayer, H., 2017. A TV Prior for High-Quality Scalable Multi-View Stereo Reconstruction. *International Journal of Computer Vision* 124(1), pp. 2–17.

Kuhn, A., Huang, H., Drauschke, M. and Mayer, H., 2016. Fast Probabilistic Fusion of 3D Point Clouds via Occupancy Grids for Scene Classification. In: *XXIII ISPRS Congress, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. III-3, p. accepted.

Lauterbach, H., Borrmann, D., Heß, R., Eck, D., Schilling, K. and Nüchter, A., 2015. Evaluation of a Backpack-Mounted 3D Mobile Scanning System. *Remote Sensing* 7(10), pp. 13753–13781.

Li, H. and Hartley, R., 2007. The 3d-3d Registration Problem Revisited. In: *International Conference on Computer Vision (ICCV)*, pp. 1–8.

Liu, R., Burschka, D. and Hirzinger, G., 2007. On the way to water-tight mesh. In: *3D-ARCH 2007: "Virtual Reconstruction and Visualization of Complex Architectures"*, *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVI-5/W47.

Magnusson, M., Lilienthal, A. and Duckett, T., 2007. Scan Registration for Autonomous Mining Vehicles Using 3D-NDT. *Journal of Field Robotics* 24(10), pp. 803–827.

Márton, Z.-C., Rusu, R. B. and Beetz, M., 2009. On Fast Surface Reconstruction Methods for Large and Noisy Point Clouds. In: *International Conference on Robotics and Automation*, pp. 3218–3223.

Michelini, M. and Mayer, H., 2016. Efficient Wide Baseline Structure from Motion. In: *XXIII ISPRS Congress, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. III-3, pp. 99–106.

Mishra, R. and Zhang, Y., 2012. A Review of Optical Imagery and Airborne LiDAR Data Registration Methods. *The Open Remote Sensing Journal* 5(1), pp. 54–63.

Munoz, D., Bagnell, J., Vandapel, N. and Hebert, M., 2009. Contextual Classification with Functional Max-Margin Markov Networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 975–982.

Nguatem, W. and Mayer, H., 2016. Contiguous Patch Segmentation in Pointclouds. In: *Pattern Recognition (GCPR) 2016, Lecture Notes in Computer Science*, Vol. 9796, pp. 131–142.

Pomerleau, F., Colas, F. and Siegwart, R., 2015. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Foundations and Trends in Robotics* 4(1), pp. 1–104.

Pomerleau, F., Colas, F., Siegwart, R. and Magnenat, S., 2013. Comparing ICP Variants on Real-World Data Sets. *Autonomous Robots* 34(3), pp. 133–148.

Rabbani, T., van Den Heuvel, F. and Vosselman, G., 2006. Segmentation of Point Clouds Using Smoothness Constraint. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVI-5, pp. 248–253.

Roth, L., Kuhn, A. and Mayer, H., 2017. Wide-Baseline Image Matching with Projective View Synthesis and Calibrated Geometric Verification. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science* 2017(2), pp. 1–11.

Rupnik, E., Pierre-Deseilligny, M. and Delorme, A., 2018. 3D Reconstruction from Multi-View VHR-Satellite Images in MicMac. *ISPRS Journal of Photogrammetry and Remote Sensing* 139, pp. 201–211.

Rusu, R. B. and Cousins, S., 2011. 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–4.

Schenk, T. and Csathó, B., 2002. Fusion of LiDAR Data and Aerial Imagery for a More Complete Surface Reconstruction. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 34 (3/A), pp. 310–317.

Vosselman, G., Coenen, M. and Rottensteiner, F., 2017. Contextual Segment-Based Classification of Airborne Laser Scanner Data. *ISPRS Journal of Photogrammetry and Remote Sensing* 128, pp. 354–371.

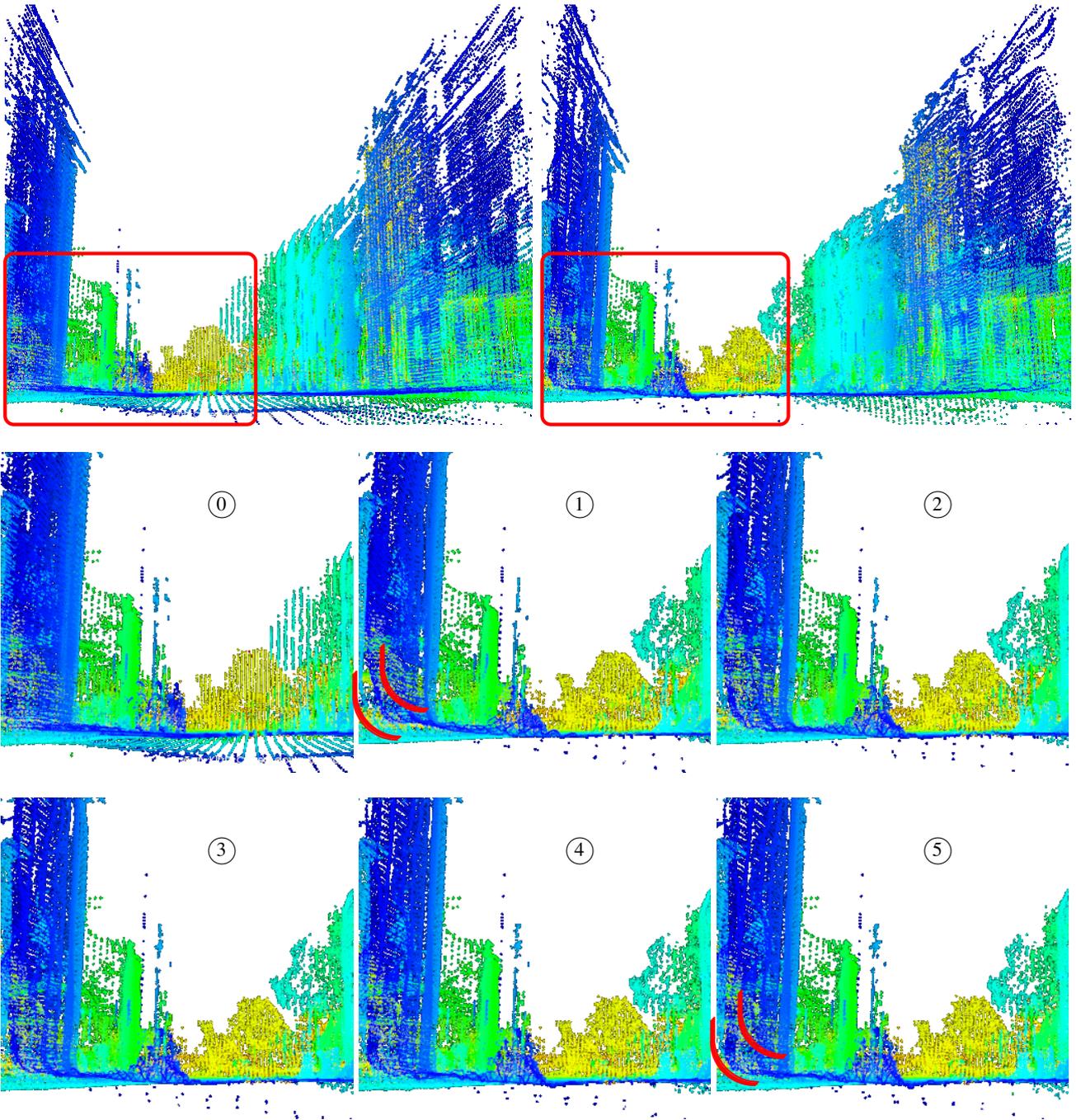


Figure 10. Top row: Result of fusion, left: with standard ICP only, right: with ICP and an additional step with surface smoothing using a polynomial of order 5. Thereto, the points from landscape model are projected on polynomial models through LiDAR points. A box is marked in both views. This extract is documented in the two rows below. There, the results without polynomial fittings (0) and with polynomial fittings of degree 1 to 5 are visualized in the according subfigures. Additionally, we marked the area of a building footprint in two views. The increased accuracy along the edge between facade and road can get inspected. Furthermore, notice the disappearance of the doubled road surface. This happens already at 1st order polynomial surface estimation.

Wagner, R., Frese, U. and Bäuml, B., 2014. Graph SLAM with Signed Distance Function Maps on a Humanoid Robot. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2691–2698.

Wang, S., Bai, M., Mattyus, G., Chu, H., Luo, W., Yang, B., Liang, J., Chaverie, J., Fidler, S. and Urtasun, R., 2017. TorontoCity: Seeing the World With a Million Eyes. In: IEEE International Conference on Computer Vision (ICCV), pp. 3009–3017.

Weinmann, M., Jutzi, B. and Mallet, C., 2014. Semantic 3d Scene

Interpretation: A Framework Combining Optimal Neighborhood Size Selection with Relevant Features. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. II-3, pp. 181–188.

Zhou, Q.-Y., Park, J. and Koltun, V., 2016. Fast Global Registration. In: European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science, Vol. 9906, pp. 766–782.