ACCURACY ASSESSMENT OF BUILDING POINT CLOUDS AUTOMATICALLY GENERATED FROM IPHONE IMAGES

B. Sirmacek, R. Lindenbergh

Delft University of Technology, Department of Geoscience and Remote Sensing, Stevinweg 1,
2628CN Delft, The Netherlands
b.sirmacek@tudelft.nl, r.c.lindenbergh@tudelft.nl

KEY WORDS: Registration, Iterative Closest Points (ICP), Point Clouds, LIDAR, Smartphone, iPhone, 3D City Models, Terrestrial Laser Scanning (TLS), Structure from Motion (SfM), Bundle Adjustment, Multi-view Photogrammetry, Low-cost sensors

ABSTRACT:

Low-cost sensor generated 3D models can be useful for quick 3D urban model updating, yet the quality of the models is questionable. In this article, we evaluate the reliability of an automatic point cloud generation method using multi-view iPhone images or an iPhone video file as an input. We register such automatically generated point cloud on a TLS point cloud of the same object to discuss accuracy, advantages and limitations of the iPhone generated point clouds. For the chosen example showcase, we have classified 1.23% of the iPhone point cloud points as outliers, and calculated the mean of the point to point distances to the TLS point cloud as 0.11m. Since a TLS point cloud might also include measurement errors and noise, we computed local noise values for the point clouds from both sources. Mean (μ) and standard deviation (σ) of roughness histograms are calculated as ($\mu_1 = 0.44m., \sigma_1 = 0.071m.$) and ($\mu_2 = 0.025m., \sigma_2 = 0.037m.$) for the iPhone and TLS point clouds respectively. Our experimental results indicate possible usage of the proposed automatic 3D model generation framework for 3D urban map updating, fusion and detail enhancing, quick and real-time change detection purposes. However, further insights should be obtained first on the circumstances that are needed to guarantee a successful point cloud generation from smartphone images.

1. INTRODUCTION

Point clouds are developing towards a standard product in urban management. Still, outdoor point cloud acquisition with active sensors is a relatively expensive and involved process. Generation of point clouds using smartphone sensors could be a rapid, cheap and less involved alternative for local point cloud generation, that could be applied for 3D archive updating or for quick damage assessment. Before smartphone generated point clouds can be integrated in an operational workflow it is essential to assess the quality of such cheap point clouds. Therefore in this study, we analyse a workflow for point cloud generation using iPhone sensors. First, we discuss how to generate a point cloud from multi-view iPhone images and from iPhone videos. Then we calculate the quality of the resulting point clouds using TLS point clouds as reference. We also discuss advantages/limitations of iPhone based point cloud generation and open questions in detail.

Modelling 3D urban structures gained popularity in urban monitoring, safety, planning, entertainment and commercial applications. 3D models are valuable especially for simulations. Most of the time models are generated from airborne or satellite sensors and the representations are improved by texture mapping. This mapping is mostly done using optical aerial or satellite images and texture mapping is applied onto 3D models of the scene (Mastin et al., 2009), (Kaminsky et al., 2009). One of the traditional solutions for local 3D data capturing is the use of a Terrestrial Laser Scanner (TLS). Unfortunately, these devices are often very expensive, require careful handling by experts and complex calibration procedures and they are designed for a restricted depth range only. On the other hand, high sampling rates with millimetre accuracy in depth and location makes TLS data a quite reliable source for acquiring measurements. Therefore, herein we use TLS data as reference to evaluate the accuracy of the iPhone point cloud.

In last years, there has been a considerable amount of research on 3D modelling of urban structures. (Liu et al., 2006) applied structure-from-motion (SFM) to a collection of photographs to infer a sparse set of 3D points, and furthermore they performed 2D to 3D registration by using camera parameters and photogrammetry techniques. Another work (Zhao et al., 2004) introduced stereo vision techniques to infer 3D structure from video sequences, followed by 3D-3D registration with the iterative closest point (ICP) algorithm. The main challenge with these methods is that they require numerous overlapping images of the scene. Some of the significant studies in this field focused into the alignment work (Huttenlocher and Ullman, 1990) and the viewpoint consistency constraint (Lowe, 2004). Those traditional methods assume a clean, correct 3D model with known contours that produce edges when projected. 2D shape to image matching is another well-explored topic in literature. The most popular methods include chamfer matching, Hausdorff matching (Huttenlocher and Kl, 1993) and shape context matching as (Belongie and Malik, 2002) introduced. Koch et al. (Koch et al., 1998) reconstructed outdoor objects in 3D by using multi-view images without calibrating the camera.

Since the last decade, some researchers focused on developing algorithms which are based on processing the images taken from smart phone sensors. (Wang, 2012) proposed a semi-automatic algorithm to reconstruct 3D building models by using images taken from smart phones with GPS and g-sensor (accelerometer) information. (Fritsch et al., 2011) used a similar idea for 3D reconstruction of historical buildings. They used multi-view smart phone images with 3D position and G-sensor information to reconstruct building facades. (Bach and Daniel, 2011) used iPhone images to generate 3D models. To do so, they also used multi-view images. They extracted building corners and edges which are used for registration and depth estimation purposes between images. After estimating the 3D building model, they have chosen one of the images for each facade with the best looking angle and they have registered that image on the 3D model for textur-

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-5-547-2014

ing it. They have provided an opportunity to the user to select their accurate image acquisition positions on a satellite map since iPhone GPS data does not always provide very accurate positioning information.

Lee and Scatto (Lee and Scatto, 2010) demonstrated a feature tracking and object recognition application using videos captured by the iPhone 3GS camera sensor. Heidori et al. (Heidari and Alaei-Novin, 2013) proposed an object tracking method using the iPhone 4 camera sensor. These studies show the usability of iPhone images for feature extraction and matching purposes which is also one of the important steps of 3D depth measurement from multi-view images. On the other hand there are some disadvantages. Unfortunately, iPhone videos have low frame rates, have a rolling shutter, and most importantly, they have a narrow field of view. Each of these effects might make iPhone image and video processing problematic (Klein and Murray, 2009). In order to cope with these physical challenges, Klein and Murray (Klein and Murray, 2009) applied a rolling shutter compensation algorithm at the Bundle adjustment stage.

In our study, we reconstruct point clouds automatically using popular computer vision algorithms. We start with acquiring iPhone sensor data either by making multi-view pictures (as introduced in (Sirmacek et al., 2013)) or by capturing a video around the object of interest. If the sensor input is a video file, then the frame rate is reduced and the frames are considered as multi-view input images. In the current stage of research, the proposed approach does not require camera calibration, however in the future we would like to consider camera calibration and lens distortion compensation as well. We discuss the accuracy of the point clouds which are generated using an iPhone sensor by using TLS point clouds as reference.

2. IPHONE AND TLS POINT CLOUDS

Herein, we first introduce methodology for data acquisition and point cloud generation using the iPhone sensor. For quantitative assessment, we use terrestrial laser scanning measurements. Fig. 1, left, shows the building of interest which is chosen as an example to explain the steps of the method. Fig. 1, right, shows a map of the building and scanner locations.

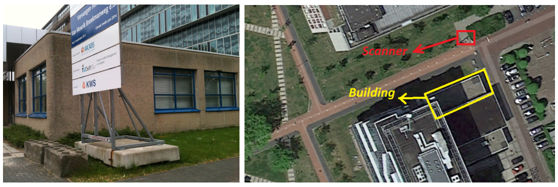

Figure 1: Left; the building chosen for our first showcase. Right; Google earth view indicating the location of the building of interest and the TLS scanner position.

2.1 iPhone Point Cloud

The iPhone point cloud is generated from iPhone 3GS smartphone sensor data. In order to generate point clouds we used multi-view images and applied a photogrammetry based method. Multi-view images are acquired in two different ways, either several photos are taken from different looking angles or a video is captured around the object of interest. If the input is a video file then the video frame rate is reduced to 10% and the frames are used as multi-view input images. For our example showcase, we use multi-view photographs. Some samples of the input photographs are shown in Fig. 2.

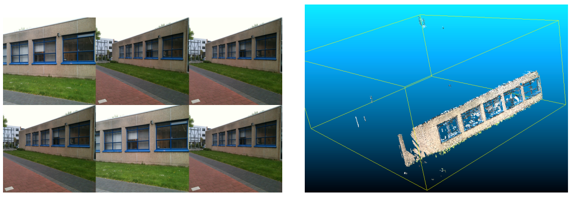

Figure 2: Left; some of the multi-view images of the building of interest. Right; point cloud generated from multi-view iPhone images.

The algorithm starts by extracting local features of each input image. In our study, we use the SIFT method for feature and descriptor vector extraction (Lowe, 2004). The smallest Euclidean distances between the descriptor vectors are considered for matching SIFT features of overlapping input images. After applying SIFT feature matching, the relative rotation, translation and position matrices are calculated and these matrices are used as input for the structure from motion (SfM) algorithm (Hartley, 1993) in order to estimate the internal and external camera parameters. These are used for initializing the bundle adjustment algorithm which helps us by calculating the complete 3D point cloud.

In Fig. 3, 4 and 5, we present some more iPhone generated point cloud examples. Fig. 3 shows a point cloud sampling of a historical windmill. For such a complex and irregular shaped structure, we find the generated point cloud satisfying for extracting information. In Fig. 4, we show a frame from video input of another showcase building at the left side, and at the right side we show the automatically generated point cloud. Unfortunately, this facade is partially occluded by trees, which strongly reduces the density of the generated point cloud. In Fig. 5, on the top image, again we show a sample frame from a video input. In the bottom we show the automatically generated point cloud which seems dense and satisfying.

2.2 TLS Point Cloud

Evaluation of the quality of iPhone generated point clouds is done by comparing them with Terrestrial Laser Scanner (TLS) point clouds. In this study, for reference point cloud generation, we use a *FARO Photon 120/20* laser scanner which is capable of gathering up to 976000 points-per-second at a maximum distance of 100 meters with an accuracy of 2mm ranging error at 25 m (Faro, 2014). We show our scanner in Fig. 6.

Each captured 3D point is associated with four values (x, y, z, l) where (x, y, z) are its Cartesian coordinates in the scanner's local coordinate system, and l is the laser intensity of the returned laser beam. This scan takes approximately one minute. In Fig. 7, we show a view of the 3D point cloud for the example showcase. As it can be seen, the occlusion limits the data acquisition from the front facade and the looking angle to the left wall causes less dense point cloud generation.

3. COMPARING THE POINT CLOUDS

In order to compare iPhone and TLS point clouds, the point clouds must be registered to each other. Since both point clouds contain outliers, first we start with eliminating the outliers and leave only the points of interest in the point cloud data. To do so, we apply a pre-processing step based on connected component analysis in 3D space which is performed by octree segmentation. We assume that the largest connected component must contain the points belonging to our object of interest (i.e. the building facade). Therefore, we keep the points of the largest segment and eliminate all

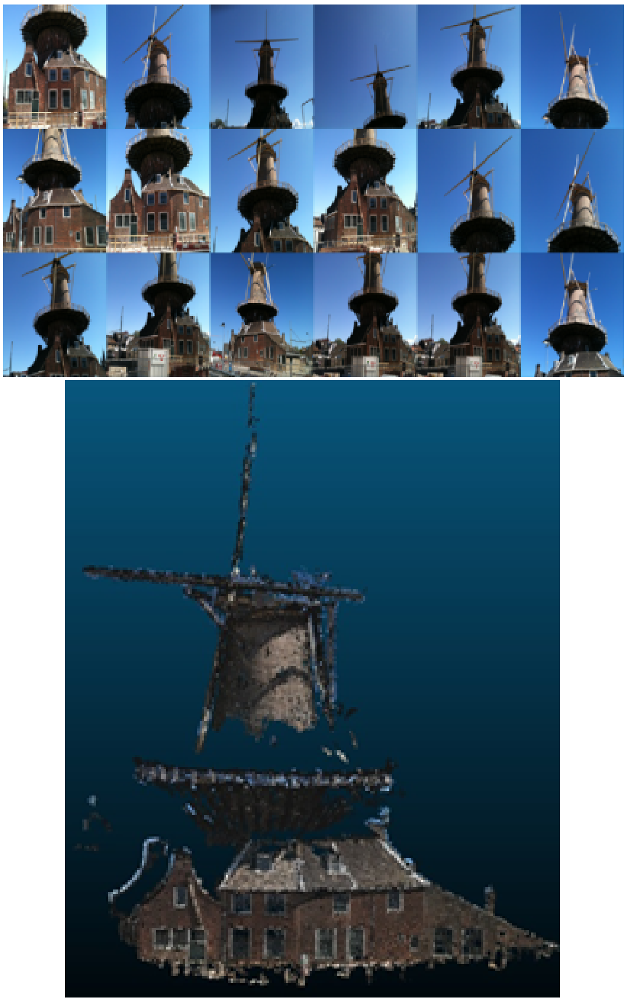

Figure 3: Top; Some samples from multi-view iPhone image data set of the historical windmill. Bottom; Automacatically generated point cloud.

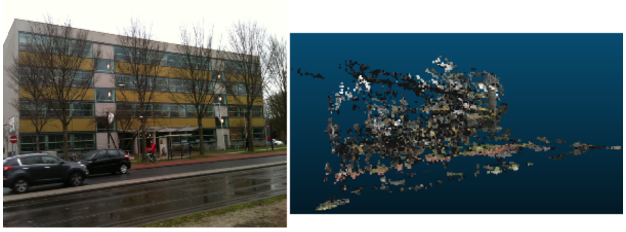

Figure 4: Left; A sample iPhone video frame from the building of interest. Right; Automacatically generated point cloud.

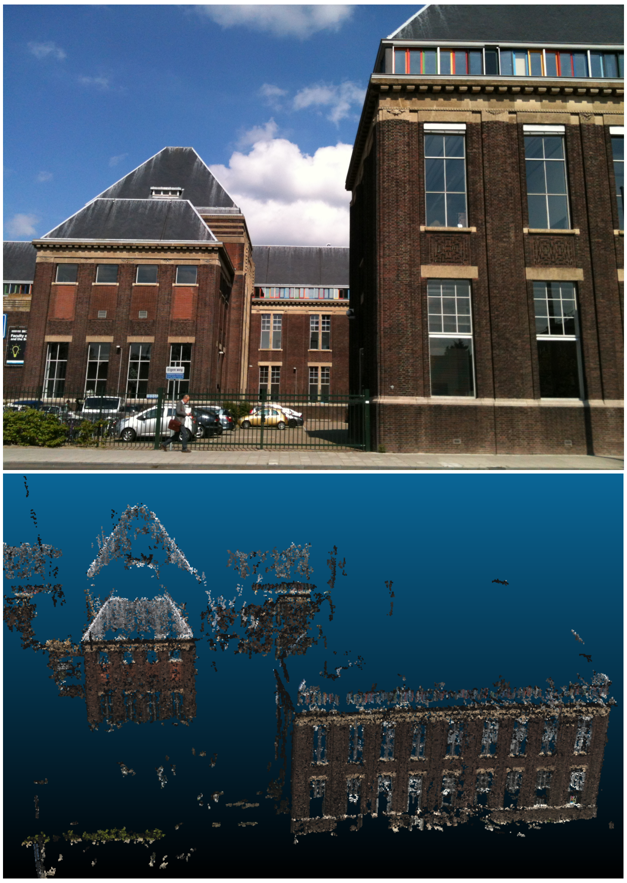

Figure 5: Top; A sample iPhone video frame from the building of interest. Bottom; Automacatically generated point cloud.

Figure 6: FARO Photon 120/20 model scanner.

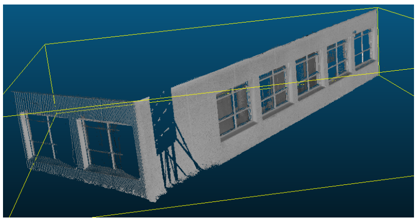

Figure 7: The point cloud obtained by our TLS device.

other points from the point cloud. In Fig. 8, we show the detected segments based on connected component analysis and the selected segment for iPhone point cloud analysis. In Fig. 9, we show the detected segments for the TLS point cloud after outlier removal.

To register the point clouds, we first start with coarsely aligning them. Since iPhone images are not georeferenced nor calibrated, their point cloud appear in an arbitrary place in the 3D space, in an arbitrary orientation and scale. Especially when the scale is different most of the commonly used registration algorithms cannot reach a result. Therefore, we first scale the point cloud manually by using a point cloud processing software and again with the same software we coarsely align the iPhone point cloud on the TLS point cloud. Then, we start the point cloud registration process. One of the most well-known algorithms for registering two

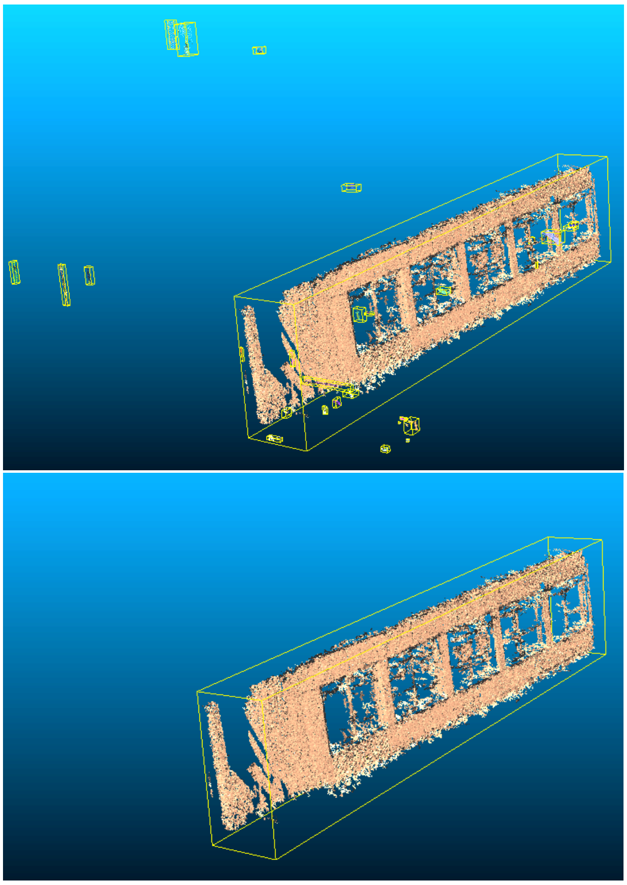

Figure 8: Top; segments obtained by connected component analysis. Bottom; Largest segment is selected as the object of interest and the other segments are removed.

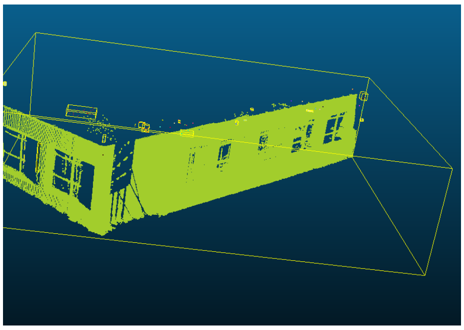

Figure 9: Left; segments obtained by connected component analysis. Right; Largest segment is selected as the object of interest and the other segments are removed.

point clouds based on overlapping surfaces is the iterative closest point (ICP) method (Besl and Mckay, 1992), an iterative algorithm which requires an initial rough registration. Essentially, the algorithm steps are as follows;

- For each point in the source (iPhone) point cloud, find the closest point in the reference (TLS) point cloud.
- Estimate the combination of rotation and translation using a mean squared error cost function that will best align each source point to its match found in the previous step.
- Transform the source points using the obtained transformation function.
- Iterate (re-associate the points, and so on).

In Fig. 10, we represent the registration result for the showcase iPhone and TLS point clouds. In the following section, we discuss the accuracy of the results in detail.

4. ACCURACY TEST ON THE SHOWCASES

For the showcase point cloud, the iPhone point cloud is obtained by using 25 iPhone photographs taken from different positions. The resulting point cloud contains 230876 points. The number of points might increase if more input images are used, however that also leads to higher computation time and memory requirements. After removing outliers 228023 points remain. This means that 1.23% points of the original point cloud were considered outlier. As expected the TLS point cloud for the same surface is denser. In the TLS point cloud, after removing the outliers, we have 357118 points. After registering these two point clouds, we analyse their differences by determining the point to point distances. In Fig. 10, we show the point to point distances with color codes. Here blue color shows the source points which have '0'm. distance to the reference data. Yellow and red colors show the source points which have higher distances to the reference point cloud. In Fig. 11, we present the distribution of the distances that we have calculated between iPhone and TLS point clouds. By fitting a Gaussian model to fit on this distribution, we have calculated the mean value as 0.11 m. and the standard deviation as 0.08 m.

Since the reference TLS point cloud might also contain measurement errors and noise, we determine a local roughness values for both iPhone and TLS point clouds. For roughness value computation, each point cloud is processed independently. For the point cloud at hand, a kernel size is selected by the user. The kernel size determines the radius of a sphere which is centred on each point. For each point, one roughness value is calculated inside the kernel. The roughness value is equal to the distance of the point to the least square best fitting plane which fits on the neighbouring points which are inside the kernel. For roughness evaluation of the showcase point clouds (iPhone and TLS), a kernel size is chosen as 0.5m. When we look at the histogram of the roughness values, mean (μ) and standard deviation (σ) of the roughness histograms are calculated as ($\mu_1 = 0.44m., \sigma_1 = 0.071m.$) and ($\mu_2 = 0.025m., \sigma_2 = 0.037m.$) for the iPhone and TLS point clouds respectively. These results show us that the reference TLS point cloud also contains some measurement errors and noise. Besides it shows us that the higher sigma value of the iPhone point cloud roughness histogram is caused by not only the error and noise, but also by the embossed window edges which also clearly appear on the TLS point cloud.

5. DISCUSSION AND THE FUTURE WORK

The proposed method thus offers very high flexibility in 3D model acquisition with cheap and easy-to-use sensors. Although the experiments are done using iPhone images and videos, any smartphone sensor would be sufficient for data acquisition and 3D reconstruction.

We have observed that both iPhone and TLS point clouds have their own advantages and disadvantages. Some of the important points can be listed as follows. TLS point clouds do not always contain color information. When the color texture of the facade needs to be displayed on the 3D data, then a normal camera photograph must be registered on the point cloud to obtain colors. TLS has a certain distance limit to scan objects. For instance, the FARO scanner which is used in this study can scan objects at maximum of 100m. distance. TLS has a good feature that it is not effected from weather and the illumination conditions. Even at night time it is possible to acquire 3D point clouds.

Last but not least, the sampling resolution of the TLS is very high. On the other hand, iPhone point clouds already contains the RGB color information registered to the points. The distance limit is not 100m., however it is hard to say after which distance we cannot do 3D reconstruction any more. We believe that, since the 3D reconstruction depends on local feature extraction and feature matching steps, the object must be in a distance, scale and resolution where the local characteristic object features can be extracted and matched correctly. Since it is a passive sensor, for iPhone point cloud generation good illumination conditions are necessary. IPhone point cloud generation also requires additional scaling and georeferencing steps.

Besides, we have some open questions that require further experiments. For example, for iPhone generated point clouds, we cannot say exactly what the looking-angle differences, maximum and minimum input photograph numbers, suggested distance to the object, sampling rate of the video -when it is the input image source- should be. We need to analyse effects of repetitive facade structures (which leads to mismatches of features), homogeneous surfaces (which doesn't give features for matching process) and occlusions. In our next study, we will also focus on automatic georeferencing and scaling of the iPhone point clouds. In this way, before registration process, pre-alignment of the iPhone point cloud on TLS point cloud will be done fully automatically. Furthermore, we will expand our quantitative assessments by comparing the iPhone generated point clouds which are given in Fig. 3, 4 and 5 with TLS point clouds.

ACKNOWLEDGEMENTS

This research is funded by the FP7 project IQmulus (FP7-ICT-2011-318787) a high volume fusion and analysis platform for geospatial point clouds, coverages and volumetric data set.

Revised May 2014

REFERENCES

Bach, M. and Daniel, S., 2011. Towards a fast and easy approach for building textured 3D models using smartphones. In proceedings of ISPRS Joint Workshop on 3D City Modelling and Applications.

Belongie, S. and Malik, J.and Puzicha, J., 2002. Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4), pp. 509–522.

Besl, P. J. and Mckay, N. D., 1992. A method for registration of 3d shapes. In IEEE Transactions on Pattern Analysis and Machine Intelligence 14, pp. 239–256.

Faro, 2014. Faro laser scanner photon 120/20 - technical and specification.

Fritsch, D., Khosravani, A., Cefalu, A. and Wenzel, K., 2011. Multi-sensors and multiray reconstrcution for digital preservation. In Proceedings of the Photogrammetric Week 2011 1, pp. 305–323.

Hartley, R., 1993. Euclidean reconstruction from uncalibrated views. in proceedings of the second European Workshop on Invariants, Ponta Delgada, Azores, Springer-Verlang 1, pp. 187–202.

Heidari, A. and Alaei-Novin, I., A. P., 2013. Fusion of spatial and visual information for object tracking on iphone. 2013 16th International Conference on Information Fusion (FUSION), 1, pp. 630–637.

Huttenlocher, D. and Kl, G.A.and Rucklidge, W., 1993. Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 15, pp. 850–863.

Huttenlocher, D. and Ullman, S., 1990. Recognizing solid objects by alignment with an image. International Journal of Computer Vision 5 (2), pp. 195–212.

Kaminsky, R., Snavely, N., Seitz, S. and Szeliski, R., 2009. Alignment of 3d point cloudstooverheadimages. in proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 1, pp. 67–70.

Klein, G. and Murray, D., 2009. Parallel tracking and mapping on a camera phone. In Proceedings of International Symposium on Mixed and Augmented Reality (ISMAR'09, Orlando).

Koch, R., Pollefeys, M. and van Gool L., 1998. Automatic 3d model acquisition from uncalibrated image sequences. Proceedings of Computer Graphics International B1, pp. 597–604.

Lee, T. and Scatto, S., 2010. Feature tracking and object recognition on a hand-held. 2010 9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR) 1, pp. 306.

Liu, L., Stamos, I., Yu, G., Wolberg, G. and Zokai, S., 2006. Multiview geometry for texture mapping 2d images onto 3d range data. in proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1, pp. 2293–2300.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal on Computer Vision 60(2), pp. 91–110.

Mastin, A., Kepner, J. and Fisher, J., 2009. Automaticregistration of lidar and optical images of urban scenes. in proceedings of IEEE Computer Vision and Pattern RecognitionConference 1, pp. 2639–2646.

Sirmacek, B., Lindenbergh, R. and Menenti, M., 2013. Automatic generation and registration of iphone point clouds on terrestrial laser scanning point clouds. In proceedings of the 6th ISPRS workshop on Laser Scanning.

Wang, S., 2012. Integrating sensors on a smartphone to generate texture images of 3D photo-realistic building models. Proceedings of the Global Geospatial Conference 2012, Quebec City.

Zhao, W., Nister, D. and Hsu, S., 2004. Alignment of continuous video onto 3d clouds. in proceedings of the Computer Vision and Pattern Recognition Conference 2, pp. 964–971.

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-5-547-2014

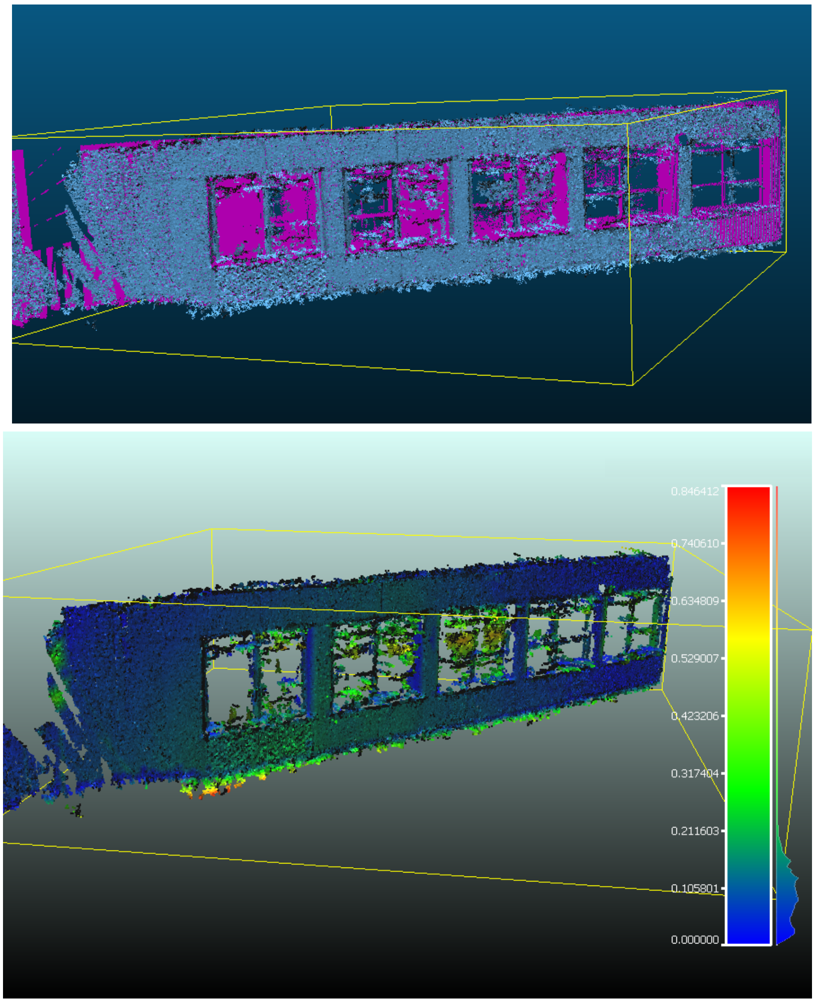

Figure 10: Top; Registered iPhone and TLS point clouds. Bottom; point to point distances are shown with pseudocoloring.

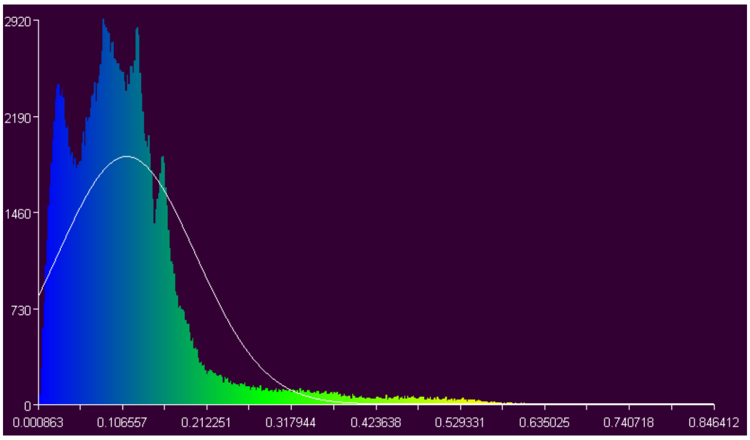

Figure 11: Distribution of the distances (in meter) between the example iPhone point cloud and TLS point cloud.