AUTOMATIC REGISTRATION OF IPHONE IMAGES TO LASER POINT CLOUDS OF URBAN STRUCTURES USING SHAPE FEATURES

Beril Sirmacek, Roderik Lindenbergh, Massimo Menenti

Department of Geoscience and Remote Sensing, Delft University of Technology, Stevinweg 1, 2628CN Delft, The Netherlands
(B.Sirmacek, R.C.Lindenbergh, M.Menenti)@tudelft.nl

Commission I/3

KEY WORDS: Feature Extraction, Feature Matching, Registration, Point Clouds, LIDAR, Mobile Mapping, Smartphone, iPhone, Texture Mapping, Facade Detection, 3D City Models, Building Modelling, Localization, GPS, Airborne Laser Scanning, Derivative Filters

ABSTRACT:

Fusion of 3D airborne laser (LIDAR) data and terrestrial optical imagery can be applied in 3D urban modeling and model up-dating. The most challenging aspect of the fusion procedure is registering the terrestrial optical images on the LIDAR point clouds. In this article, we propose an approach for registering these two different data from different sensor sources. As we use iPhone camera images which are taken in front of the interested urban structure by the application user and the high resolution LIDAR point clouds of the acquired by an airborne laser sensor. After finding the photo capturing position and orientation from the iPhone photograph metafile, we automatically select the area of interest in the point cloud and transform it into a range image which has only grayscale intensity levels according to the distance from the image acquisition position. We benefit from local features for registering the iPhone image to the generated range image. In this article, we have applied the registration process based on local feature extraction and graph matching. Finally, the registration result is used for facade texture mapping on the 3D building surface mesh which is generated from the LIDAR point cloud. Our experimental results indicate possible usage of the proposed algorithm framework for 3D urban map updating and enhancing purposes.

1 INTRODUCTION

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. As in previous studies of Mastin et al. (2009) and Kaminsky et al. (2009), this mapping is mostly done using optical aerial or satellite images and texture mapping is applied onto 3D models of the scene. 3D models are either generated by multiple view stereo images using triangulation techniques. Some of the researchers generated 3D models manually. Recently, advances in airborne laser radar (LIDAR) imaging technology have made the acquisition of high resolution digital elevation models more efficient and cost effective.

One challenge in creating realistic models is registering 2D optical imagery with the 3D LIDAR imagery. This can be formulated as a camera pose estimation problem where the transformation between 3D LIDAR coordinates and 2D image coordinates is characterized by camera parameters. Manual camera pose selection is difcult as it requires simultaneous renement of numerous camera parameters. Registration can be applied more efciently by manually selecting pairs of correspondence points, but this work might become tedious for situations where many images must be registered to create large 3D urban models. Some methods have been developed for performing automatic registration, but they suffer from being computationally expensive and/or demonstrating low accuracy rates.

In previous work, there has been a considerable amount of research in registering optical images either with LIDAR or 3D models obtained by stereo imaging. Liu et al. (2006) applied structure-from-motion (SFM) to a collection of photographs to infer a sparse set of 3D points, and then performed 2D to 3D registration by using camera parameters and photogrammetry techniques. An 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.

Classical work on object recognition includes more examples of the registration of single 2D images onto 3D models. Some of the significant studies in this field include the alignment work Huttenlocher and Ullman (1990) and the viewpoint consistency constraint Lowe (1987) matched the projections of a known 3D model to 2D edge images. 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 the literature. The most popular methods include chamfer matching, Hausdorff matching Huttenlocher et al. (1993) and shape context matching as Belongie et al. (2002) introduced. Ding et al. (2008) aligned LIDAR scans with oblique aerial imagery by detecting and matching corners, while Fruh and Zakhor (2004), Fruh and Zakhor (2001) registered aerial and ground-level scans. The dense 3D geometry used in these techniques allow for much more robust detection of geometric primitives such as edges and corners for matching. In the area of single-view registration, Vasile et al. (2006) introduced LIDAR data to derive a pseudo-intensity image with shadows for correlation with aerial imagery. Their registration procedure starts with GPS and camera line of sight information and then uses an exhaustive search over translation, scale, and lens distortion. Fruh et al. (2004) developed a similar system based on detection and alignment of line segments in the optical image and projections of line segments from the 3D image. Using a prior camera orientation with an accuracy comparable to that of a GPS and inertial navigation system (INS), they used an exhaustive search over camera position, orientation, and focal length. Their system

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
doi:10.5194/isprsannals-II-5-W2-265-2013

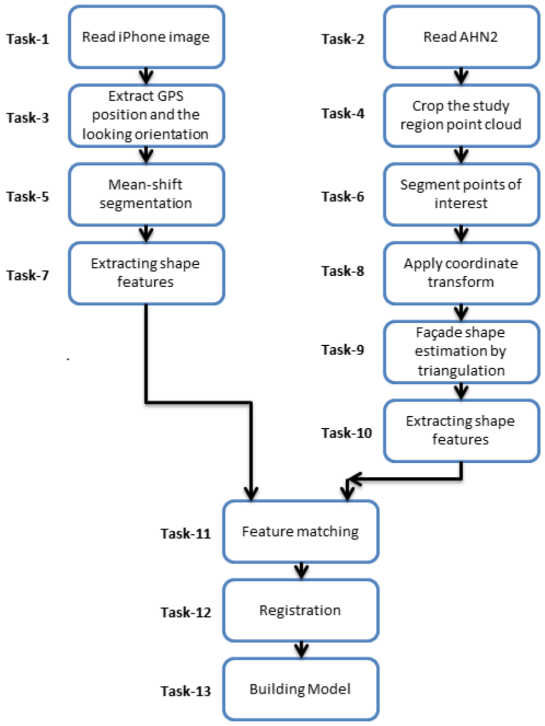

Task-1
Read iPhone image
Task-2
Read AHN2
Task-3
Extract GPS position and the looking orientation
Task-4
Crop the study region point cloud
Task-5
Mean-shift segmentation
Task-6
Segment points of interest
Task-7
Extracting shape features
Task-8
Apply coordinate transform
Task-9
Façade shape estimation by triangulation
Task-10
Extracting shape features
Task-11
Feature matching
Task-12
Registration
Task-13
Building Model

Figure 1: The proposed work flow chart.

requires approximately 20 hours of computing time on a standard computer. Although those methods demonstrate accurate registration results, they are computationally expensive. There are a variety of algorithms that utilize specific image features to perform registration. Troccoli and Allen (2004) used matching of shadows to align images with a 3D model. This requires a strong presence of shadows as well as knowledge of the relative sun position when the photographs were taken. Kurazume et al. (2005) used detection of and matching of edges for registration. Unfortunately, this method requires dense 3D point clouds to infer edges. Stamos and Allen (2002) used matching of rectangles from building facades for alignment. Yang et al. (2007) used feature matching to align ground images. These methods are not robust for all types of urban imagery, and are not optimal for sparse point clouds. Some of the other approaches have employed vanishing points. Lee et al. (2002) extracted lines from images and 3D models to find vanishing points. Their system cannot register all types of imagery, as it was designed for ground-based images with clearly visible facades.Ding et al. (2008) used vanishing points with aerial imagery to detect corners in a similar manner, and used M-estimator sample consensus to identify corner matches. Starting with a GPS/INS prior, their algorithm runs in approximately 3 minutes, but only achieves a 61$\%$ accuracy rate for images of a downtown district, a college campus, and a residential region. Liu and Stamos (2007) used vanishing points and matching of features to align ground images with 3D range models. All of these approaches are dependent on the strong presence of parallel lines to infer vanishing points which limits their ability to handle different types of imagery. Since at the last decade smart phone based applications started to become more popular, 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 information. Fritsch et al. (2011) used a similar idea for 3D reconstruction of the 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. They have provided an opportunity to the user to select their accurate image acquisition positions on the satellite map since iPhone GPS data does not always provide very accurate position.

To the best of our knowledge, in the current literature, fully automatic registration of 2D terrestrial data onto 3D models generated by airborne sensors which have very low looking side overlap is so far not considered. In this article, we propose a system for this case and we represent a possible case story on a sample data set including an iPhone image and LIDAR point cloud of an urban structure. In Fig. 1, we represent our work flow chart that we have used in this study. The tasks numbers next to the flow chart steps will be referred in the rest of the article to reduce the complexity of the framework description.

2 DATA ACQUISITION AND PREPROCESSING

In our study, we use iPhone photographs for registering texture on the 3D urban models which can be used for updating maps. An iPhone photograph can be read with its metafile which is written in exchangeable image file format (Exif). Exif is a standard that specifies the formats for images, sound and other digital records like videos or scanner data. The metafile contains a wide spectrum of tags like, date and time information, camera properties, GPS position, looking angle, image resolution and properties, device properties, etc. Reading the iPhone image and its Exif data is represented as Task-1 in the work flow represented in Fig. 1.

As airborne LIDAR point cloud source, we use data from the Dutch airborne laser altimetry archive Actual Height model of the Netherlands (AHN). The first AHN data was acquired between 1996 and 2003 under leaf-off conditions with a point density of at least 1 point per $4 \times 4m^2$ area. Starting from 2007, AHN2 is being acquired over The Netherlands as it is introduced by AHN (2008), Duong et al. (2009), Swart (2010) and van der Sande et al. (2010). Fig. 2 represents AHN data of the Netherlands as completed in 2003. Since AHN2 has higher point density per square meter, in this study we use old city center of Delft city Netherlands AHN2 LIDAR point cloud as data. Reading the AHN2 point cloud of the interest area is represented as Task-2 in the work flow in Fig. 1.

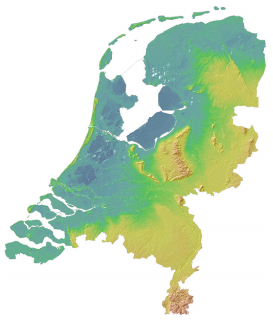


Figure 2: AHN data of the Netherlands which was completed in 2003.

The iPhone metafile gives GPS geolocations and photographing angle. This provides us opportunity to find our location in the point cloud. Accessing the geolocations and photographing angle is represented as Task-3 in the work flow represented in Fig. 1. Since we know our study region more precisely, as Task-4 we crop the LASER point cloud to reduce the data to process. Our crop function creates a subset of the point cloud by collecting the closest 10000×10000 points around the iPhone image acquisition position as described in iPhone metafile.

3 SHAPE FEATURES FOR DATA REGISTRATION

3.1 Extracting Shape Features from iPhone Data

For shape feature extraction, iPhone photographs include many details of the objects and their textures which give challenges to extract representative shape features. In order to decrease the complexity of the problem, Sirmacek (2011) used the mean shift segmentation algorithm to simplify the object appearances in the photographs. We apply mean shift segmentation to the iPhone image ($I(x, y)$) image as Comanicu and Meer (2002) proposed. At mean shift segmentation, we chose the spatial bandwidth (h_s) and spectral bandwidth (h_r) parameters as 7 and 6.5 respectively after extensive tests. The segmentation result is a new image denoted $S(x, y)$ which holds each segment labeled by a different number. We provide the mean shift segmentation result of our iPhone test image in Fig. 5. Unfortunately, the shapes of the segmented objects still contain many high resolution details which increases the complexity. To overcome this problem, we apply nonlinear $[7 \times 7]$ pixel size median filtering to smooth the details of the $S(x, y)$ segmentation result. The filter response is stored in image $S_f(x, y)$. As in previous studies of Sirmacek and Unsalan (2011), here we also benefit from such nonlinear smoothing operations to decrease complexity of feature extraction problems. Obtaining $S_f(x, y)$ corresponds to the Task-5 in work flow in Fig. 1.

To extract these shape features, we use a steerable filter set on the smoothed segmentation result $S_f(x, y)$. Then, extracted features help us to find the similarity between the AHN2 appearance of the building for the registration purposes in the further steps of our algorithm framework. Our shape feature extraction works similar to object detection study of Sirmacek and Unsalan (2012) which focus on the detection of the buildings from remotely sensed optical images. As proposed by Orrite et al. (1999), edges and curvilinear shapes are crucial features to identify objects in remotely sensed images. In order to extract shape features of the object segments, herein we apply steerable filters in different orientations. For a symmetric Gaussian function $G(x, y) = exp(-(x^2 + y^2))$, it is possible to define basis filters Gp_0 and $Gp_{\frac{\pi}{2}}$ as

$$Gp_0 = \frac{\partial}{\partial x} G(x, y) = -2x \exp(-(x^2 + y^2)) \tag{1}$$

$$Gp_{\frac{\pi}{2}} = \frac{\partial}{\partial y} G(x, y) = -2y \exp(-(x^2 + y^2)) \tag{2}$$

We find a derivative in an arbitrary direction θ using the following rotation

$$Gp_\theta = cos(\theta) Gp_0 + sin(\theta) Gp_{\frac{\pi}{2}} \tag{3}$$

After obtaining a steerable filter function in the θ direction, we convolve $S_f(x, y)$ with filter Gp_θ ($J_\theta(x, y) = S_f(x, y) * Gp_\theta$), to detect structural features in the θ direction. In $J_\theta(x, y)$, we expect to obtain high responses on structures which are perpendicular to the filtering direction. Therefore, we obtain our shape features by thresholding $J_\theta(x, y)$. We pick the threshold value as 20% of the maximum magnitude in $J_\theta(x, y)$ after extensive testing. After thresholding $J_\theta(x, y)$, we obtain a binary image $B_\theta(x, y)$ with pixel locations having a value of one when representing a shape feature. As it is introduced by Sonka et al. (1999), we assume each connected pixel group as one shape feature. We expect this shape extraction method to help us for robust object identification as in studies of Sirmacek and Unsalan (2012). We extract structural features in a set of θ directions. In this study, we pick our steerable filtering directions as $\theta \in \{0, \pi/4, \pi/2, 3 * \pi/4\}$. The extracted shape features for our example iPhone image is represented in Fig. 6. After this shape feature extraction operation, we may have either a straight line segments or L shaped curves in $B_\theta(x, y)$ $\theta \in \{0, \pi/4, \pi/2, 3*\pi/4\}$ binary images. The extracted iPhone photograph shape feature results are shown in Fig. 6. Using extracted shape features, we generate a graph network to understand the spatial relationships of the shape features between each other. To do so, we consider the mass centers of the shape features as nodes (V_I), and the Euclidean distances between them is considered as the edges of the graph network (E_I). A $G_I = (V_I, E_I)$ graph network is generated for the local features extracted from the iPhone image.

3.2 Extracting Shape Features from The Point Cloud Data

By using the (x_p, y_p, z_p) geographical positions and the θ_p looking angle of the iPhone camera which is read from the metafile, we extract interest points from LIDAR to be used in further processing (Task-6 of the flow chart in Fig. 1). To do so, we set search looking angles as $[\theta_p - \gamma_x, \theta_p + \gamma_x]$ and $[-\gamma_z, \gamma_z]$ from the (x_p, y_p, z_p) position where the iPhone image is captured. A previously defined constant distance away from the (x_p, y_p, z_p) position, we insert a virtual plane as in Fig. 3.(a). This plane stands between (x_p, y_p, z_p) and the LIDAR points of the building. The normal angle of the plane is in the opposite direction of the θ_p looking angle. This normal vector of the plane is illustrated in Fig. 3.(a). First, we start with a coordinate transformation to reduce the complexity of the task. We transfer interest points of LIDAR to the new coordinate system where plane normal vector represents one of the axes. After that, each point is projected on the virtual plane with a value which is equal to the perpendicular distance between point and the plane. If more than one cloud point is projected on the same position in the plane, only the point with the closest distance to the plane is kept. In this way, we perform projection only for the facade and roof points of the building which are the closest to the virtual plane. Due to the perpendicular looking angle of the airborne LIDAR sensor, unfortunately we have very sparse distribution of points sampling the building facade. In Fig. 3.(b), we present the LIDAR points which are projected on the virtual plane. Here, the red border around the points show the extracted alpha shape which is introduced by Edelsbrunner et al. (1983). In this study, we have chosen α value as 50, considering the approximate building point cloud scale. However robustness of the value needs to be analyzed further. The border points which appear on the alpha shape are checked one by one in order to decide if they can represent a discriminative feature. If the point is connected to alpha shape edges having inner angle ϕ less than a previously defined threshold ϕ_{thresh}, the point is selected as a feature. If ϕ is greater than 90 degrees, it is updated by using the equation; $\phi = \phi - 90$. The detected features are shown in Fig. 3.(a) with blue circular labels. As it can be seen in the figure, the features are extracted from sharp corners of the alpha shape. In our study, we have selected ϕ_{thresh} as 60 degrees.

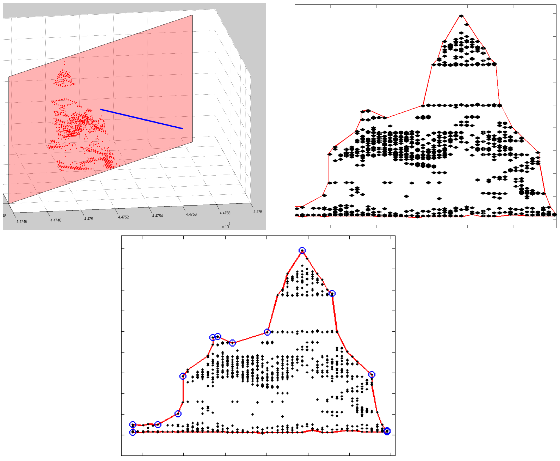


Figure 3: (a) LIDAR points of interest, virtual projection plane and its normal vector, (b) Projected LIDAR points and the extracted alpha shape, (c) Detected LIDAR features.

4 REGISTRATION OF THE I-PHONE IMAGES AND BUILDING OBTAINED FROM THE POINT CLOUD DATA

In our application, we benefit from graph theory to match features and to apply registration between the iPhone image and the projected point cloud. By using the structural feature graph $G_I = (V_I, E_I)$ which is extracted from the iPhone image and the $G_L = (V_L, E_L)$ graph which is generated using the features of the projected LIDAR points, we apply graph matching using the framework represented by the Algorithm 1.

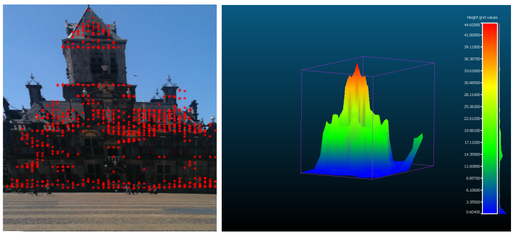


Figure 4: (a) Iphone image is registered with projected LIDAR points, (b) Generated 3D mesh model of the interest building.

In the given pseudo code, N_I and N_L represent the total structural feature number. The features are extracted from the iPhone image and the projected LIDAR points respectively. $E_I(i)$ represents the edge length in the G_I graph between the $(i-1)$th and the ith structural feature. Likewise, $E_L(j)$ represents the edge length in G_L graph between the $(j-1)$th and the jth structural feature. After finding the matching features, we use them to solve the affine transformation function for applying registration. Registration of an image to a given surface (virtual plane surface in our case), by using features to solve affine transformation parameters is explained by Moradi et al. (2006). Fig. 4.(a), shows the registered iPhone image with the projected LIDAR points of the facade. In Fig. 4.(b) represents the surface mesh which is generated by the LIDAR points of the interest object. The remaining steps of the algorithm will be using the registered iPhone image to add texture on the mesh data.

5 CONCLUSIONS AND FUTURE WORK

Algorithm 1 Matching the iPhone image features to the projected point cloud features.

```
for $i \leftarrow 1, N_I$ do
    for $j \leftarrow 1, N_L$ do
        Apply correlation between the $i$th and the $j$th structural feature
        if similarity $<$ similarity threshold then
            if $(i > 1) \wedge (j > 1)$ then
                Check if $E_I(i)$ is similar to $E_L(j)$, $len_{dif} = |E_I(i) - E_L(j)|$
                if $(len_{dif}$ $<$ length difference threshold $) \wedge ([i-1]$th feature is already matched with the $[j-1]th feature)$ then
                    $E_I(i)$ graph edge matches with $E_L(j)$ graph edge
                end if
            end if
        end if
    end for
end for
```

Herein we propose an algorithmic framework for automatic registration of iPhone images on 3D building models which are generated from airborne LASER scanner point clouds. We have shown results from initial experiments to illustrate the proposed framework by using an iPhone image and LIDAR data which belongs to the old city hall of the Delft city in the Netherlands. We hope that the proposed approach can be a novel step in the related literature, in order to add up-to-date information into existing 3D models, which can either show a new state of the urban object or makes updates on the structure if there is a significant change. In this way, it might be possible to update information and allow end-users to make contribution to the existing data sets. In order to increase the possibilities, as next steps, we would like to extract more accurate reference data to test the accuracy of iPhone GPS and orientation inputs. We will also focus on more accurate registration of the iPhone images on 3D mesh models.

ACKNOWLEDGEMENT

This research is founded 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.

References

AHN, 2008. Actual height model of The Netherlands. http://www.ahn.nl/english.php.

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., 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.

Comanicu, D. and Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5), pp. 603–619.

Ding, M., Lynbaek, K. and Zakhor, A., 2008. Automatic registration of aerial imagery with untextured 3D LIDAR models. In Proceedings of Computer Vision and Pattern Recognition Conference 1, pp. 1–8.

Duong, H., Lindenbergh, R., Pfeifer, N. and Vosselman, G., 2009. ICESat full-waveform altimetry compared to airborne LASER scanning altimetry over The Netherlands. IEEE Transactions on Geoscience and Remote Sensing 47 (10), pp. 3365–3378.

Edelsbrunner, H., Kirkpatrick, D. and Seidel, R., 1983. On the shape of a set of points in the plane. IEEE Transactions on Information Theory 29 (4), pp. 551–559.

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.

Fruh, C. and Zakhor, A., 2001. 3D model generation for cities using aerial photographs and ground level LASER scans. In Proceedings of the IEEE Computer Vision and Pattern Recognition 2, pp. 31–38.

Fruh, C. and Zakhor, A., 2004. An automated method for large scale, ground-based city model acquisition. International Journal of Computer Vision 60 (1), pp. 5–24.

Fruh, C., Sammon, R. and Zakhor, A., 2004. Automatic texture mapping of 3D city models with oblique aerial imagery. In Proceedings of 2nd International Symposium on 3D Data Processing, Visualization and Transmission pp. 396–403.

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

Huttenlocher, D., 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.

Kaminsky, R., Snavely, N., Seitz, S. and Szeliski, R., 2009. Alignment of 3D point clouds to overhead images. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops 1, pp. 67–70.

Kurazume, R., Nishino, K., Wheeler, M. and Ikeuchi, K., 2005. Mapping textures on 3D geometric model using reflectance image. Journal Systems and Computers in Japan 36 (13), pp. 92–101.

Lee, S., Jung, S. and Nevatia, R., 2002. Automatic integration of facade textures into 3D building models with a projective geometry based line clustering. Computer Graphics Forum 21 (3), pp. –.

Liu, L. and Stamos, I., 2007. A systematic approach for 2D image to 3D range registration in urban environments. In Proceedings of the IEEE ICCV pp. 1–8.

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., 1987. The viewpoint consistency constraint. International Journal of Computer Vision 1 (1), pp. 57–72.

Mastin, A., Kepner, J. and Fisher, J., 2009. Automatic registration of LIDAR and optical images of urban scenes. In proceedings of IEEE Computer Vision and Pattern Recognition Conference 1, pp. 2639–2646.

Moradi, M., Abolmaesoumi, P. and Mousavi, P., 2006. Deformable registration using scale space keypoints. In Proceedings of SPIE, Medical Imaging: Image Processing.

Orrite, C., Roy, A. and Alcolea, A., 1999. Surface segmentation based on perceptual grouping. In Proceedings of the International Conference on Image Analysis and Processing pp. 328–333.

Sirmacek, B., 2011. Graph theory and mean shift segmentation based classification of building facades. In Proceedings of the Joint Urban Remote Sensing Event (JURSE'11).

Sirmacek, B. and Unsalan, C., 2011. A probabilistic framework to detect buildings in aerial and satellite images. IEEE Transactions on Geoscience and Remote Sensing 4 (21), pp. 211–221.

Sirmacek, B. and Unsalan, C., 2012. Road network detection using probabilistic and graph theoretical methods. IEEE Transactions on Geoscience and Remote Sensing 50 (11), pp. 4441–4453.

Sonka, M., Hlavac, V. and Boyle, R., 1999. Image processing, analysis and machine vision. Second edn, PWS Publications, Pacific Grove, CA.

Stamos, I. and Allen, P. K., 2002. Geometry and texture recovery of scenes of large scale. Computer vision and image understanding 88 (2), pp. 94–118.

Swart, L., 2010. How the up-to-date height model of The Netherlands (AHN) became a massive point data cloud. in: Management of massive point cloud data: wet and dry, Nederlandse Commissie voor Geodesie, Delft.

Troccoli, A. and Allen, P., 2004. A shadow based method for image to model registration. In Proceedings of 2nd IEEE Workshop on Image and Video Registration.

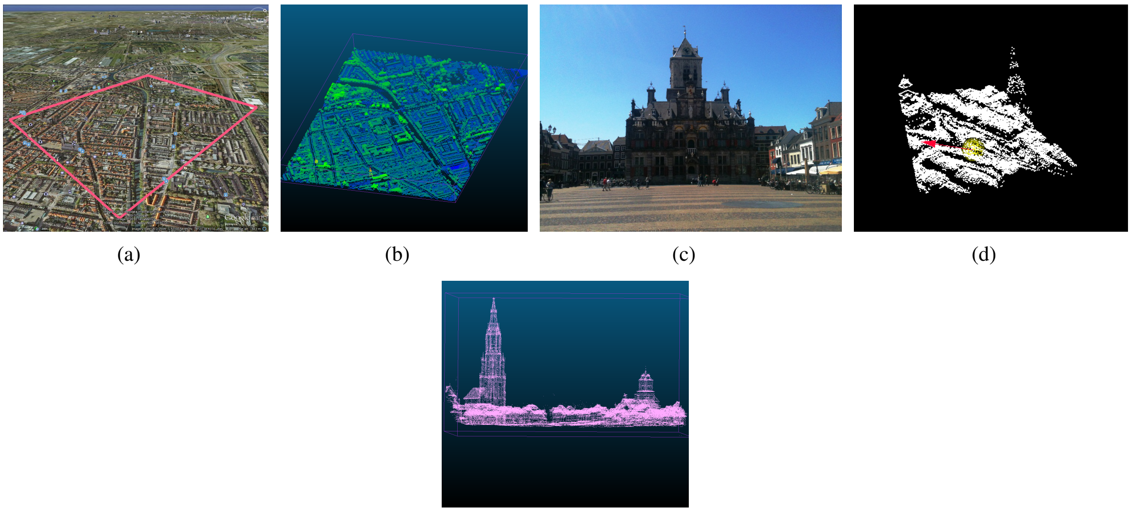
van der Sande, C., Soudarissanane, S. and Khoshelham, K., 2010. Assessment of relative accuracy of AHN2 LASER scanning data using planar features. Sensors 2010 10 (9), pp. 8198–8214.

Vasile, A., Waugh, F., Greisokh, D. and Heinrichs, R., 2006. Automatic alignment of color imagery onto 3D laser radar data. In Proceedings of the 35th Applied Imagery and Pattern Recognition Workshop.

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

Yang, G., Becker, J. and Stewart, C., 2007. Estimating the location of a camera with respect to a 3D model. In Proceedings of the International Conference on Recent Advances in 3D Digital Imaging and Modeling (3DIM) pp. 159–166.

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.



(e)

Figure 5: (a) AHN2 data acquisition area boundaries which is used in this study represented on Google Earth, (b) AHN2 point cloud height map is represented with color code, (c) iPhone photograph of the test building which is used at our demo application, (d) Geolocations of iPhone data acquisition position and the looking angle are represented on the AHN2 point cloud, (e) Sub-section of the AHN2 point cloud after selecting the interest study region considering the iPhone geolocations.

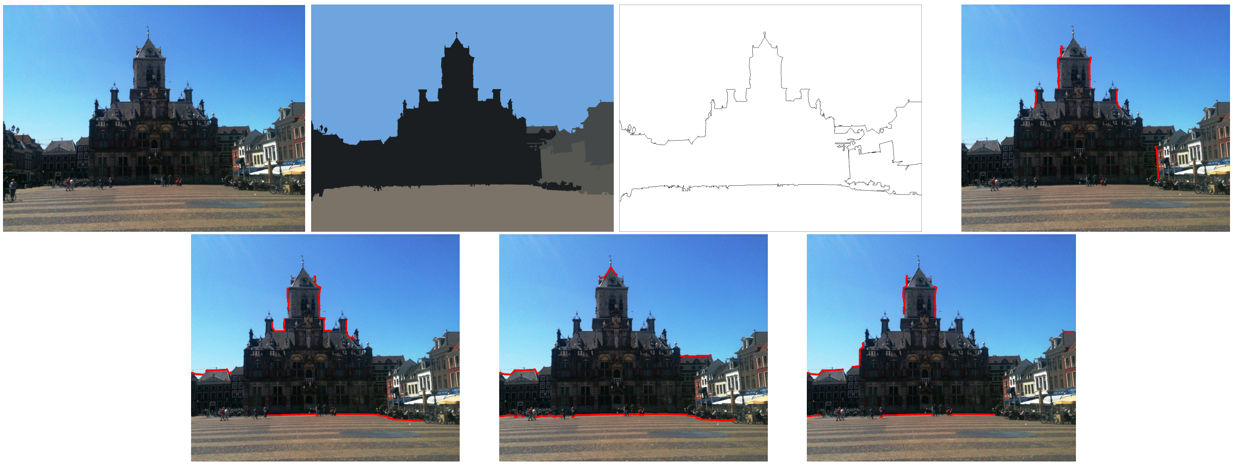


Figure 6: (a) Original iPhone photo of the interested building, (b) Mean-shift segmentation result, (c) Borders of the object segments, (d) Shape features obtained at $\theta = 0$ filtering direction, (e) Shape features obtained at $\theta = \pi/4$ filtering direction, (f) Shape features obtained at $\theta = \pi/2$ filtering direction, (g) Shape features obtained at $\theta = 3\pi/4$ filtering direction.