The Difficulty in Generating 3D City Models from Point Clouds of Dense Image Matching in Taiwan

Yen-Ting Lee[1] and Shue-Chia Wang[2]

[1]NCKU Research & Development Foundation,
No. 1, University Road, Tainan 70101, Taiwan, leeyt1377@gmail.com
[2]NCKU Research & Development Foundation,
No. 1, University Road, Tainan 70101, Taiwan, scwang@mail.ncku.edu.tw

KEY WORDS: Dense Image Matching, 3D Models, CityGML

ABSTRACT: During the last decade, dense image matching techniques have been well-developed. The high density matched point clouds are very useful for generation of various photogrammetric products like high resolution digital surface models, true-ortho images, 3D city models, and so on. Owing to the demand from urban planning, 3D cadasters, environmental protection, cell phone communications, and so forth, many countries have started to manually generate 3D city models in the 1990's with quite different specifications and formats. In order to facilitate spatial information interoperability, the Open Geospatial Consortium (OGC) adopted CityGML, developed by the Initiative Geodata Infrastructure North-Rhine Westphalia, Germany, as the international 3D city model coding standard in 2008. Nowadays, a lot of software and algorithms can extract 3D vectors of roof edges semi-automatically from dense matching's result. In this way, the cost of production 3D city models can be reduced. However, dense point clouds at the building edges and in areas with break lines, shadow and occlusions are either prone to blunders or have voids which affect the quality of the 3D models generated. Most building tops in the urban area of Taiwan are very complex so that exacting 3D roof edge vectors comply with the CityGML standard from point clouds is very difficult. This study analyzes the problems encountered during generation of 3D models according to CityGML from dense point clouds in urban area of Taiwan.

1. INTRODUCTION

Owing to the needs of urban planning, 3D cadasters, environmental protection, and cell phone communications, etc., researches on 3D city models started around the late of the 1990s(Fuchs *et al*., 1998). Today, most major cities in Japan and Europe have completed their basic 3D city models and are entering into the subsequent phase of maintenance and improvement. However, those completed 3D city models are generated in the time where there was no international standard so that they are difficult to be shared with outsiders. Therefore, after a series of selection and evaluation, the Open Geospatial Consortium (OGC) adopted CityGML, developed by the Initiative Geodata Infrastructure North-Rhine Westphalia, Germany, as the international 3D city model coding standard in 2008(OGC, 2008) and the latest version CityGML 2.0 was published in 2012(OGC, 2012).

Traditionally, 3D models were generated manually by photogrammetric stereo measurement of each single roof which is very costly. To improve the efficiency of reconstruction of 3D buildings, a lot of studies have tried to develop automatic techniques for this task. For example, with the development of airborne light detection and ranging (LiDAR), Haala and Brenner(1999) used 2D ground plans and a laser DSM to derive 3D building geometry automatically. Brunn and Weidner(1997) separated building and vegetation areas by using of height and differential geometric information and applied parametric and prismatic building models to reconstruct them. Recently the development of the dense image matching(DIM) technique, such as semi-global matching(SGM) developed by Hirschmüller(2005), multi-view stereo(MVS) matching(Seitz *et al*., 2006), patch-match(Bleyer *et al*., 2011) etc., can provide very dense surface point clouds almost one point per pixel. The main issue in the automation is how to derive 3D scene or objects from those dense point clouds. Major approaches can be represented by voxels (Fromherz and Bichsel, 1995; Kutulakos and Seitz, 2000), level-sets, polygon meshes(Furukawa and Ponce, 2010), or depth maps.

Even though there are a lot of methods which can reconstruct 3D city model from dense point clouds, but most of the automatically reconstructed 3D models do not comply with the CityGML standard due to the above mentioned deficiency in areas with break lines, shadow and occlusion. Besides, CityGML demands that each planar surface of a building should be reconstructed as one single plane only. But most planar surfaces reconstructed from point clouds are resolved in a number of irregular triangular meshes. In case of buildings with complex geometry (Figure 1) the number of planar surfaces reconstructed needs to be heavily reduced in order to let users and software operate that 3D models effectively. Schnabel *et al*.,(2007) presented a method based on RANSAC to simplify the composition of models and the surfaces of model are composed of basic shapes only. Figure 1 shows some examples of the complexity of roofs that might lead to erroneous and incomplete reconstruction of 3D models from point clouds.

Figure 1. Various decorations built on the tops of houses in Taiwan(source: Google Maps)

2. METHODOLOGY

In this section, we will discuss CityGML standard and how to generate the 3D buildings models according to CityGML.

2.1 CityGML Standard

Figure 2 shows the five levels of detail (LoD) defined by CityGML 2.0. A DTM model generated by regularly grid or triangulated irregular network (TIN) can be categorized as LoD0. LoD1 requires that buildings above the ground are modeled as simple blocks without detailed description of the shape of roof. The next higher level of LoD2 must contain detailed models of roofs, walls, ground surfaces, and building installations, indicating that LoD2 building models have more exterior information of house structure. To collect those required information like roof edge, balcony, projection of wall surface, arcade and so on, it is necessary to combine photogrammetric stereo measurement with terrestrial LiDAR scanning. For LoD3, the detailed structures of roof and wall such as doors and windows have to be reconstructed. The highest LoD4 model must have interior structures with is beyond the scope of traditional surveying technique.

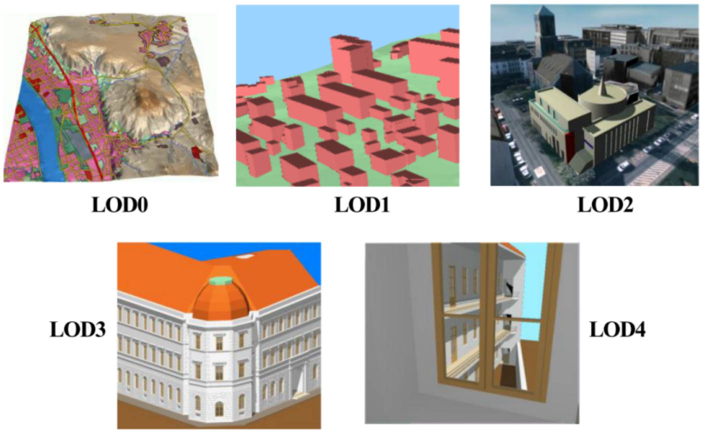


Figure 2. Five Levels of Detail for buildings in CityGML 2.0 (source: IGG Uni Bonn)

2.2 Dense Image Matching and Reconstruction of 3D Building Models

Nowadays, many commercial software and algorithms are available for generating dense point clouds using DIM technique. And there are also many software systems which reconstruct 3D building models from the point clouds. For example, SURE(Surface Reconstruction) developed by nFrames company (http://www.nframes.com/), ContexCapture developed by acute3D company (https://www.acute3d.com/contextcapture/), Correlator3D developed by Simactive company (http://www.simactive.com/en) includes a tool which can extract 3D features vectors of rooftops semi-automatically, Pointfuse (http://pointfuse.com/) converts point clouds to vector models, and so on.

In this study, we use SURE to generate dense point and Pointfuse to obtain the 3D vectors of roofs. SURE is an API software and its matching algorithm is based on SGM. It can generate dense 3D point clouds, high resolution digital surface model (DSM), mesh DSM, and ortho-image. The input data contains imagery and its orientation information and acceptable imagery includes aerial, oblique, unmanned aerial vehicle (UAV) and close range. Pointfuse is also an API software and it can reconstruct 3D models including vectors and planes automatically, like Figure 3 shows. The input data is point clouds and the output data contains various formats of vector files like DXF, OBJ, IFC ,etc..

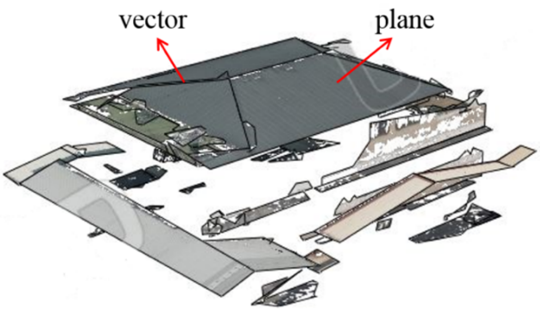


Figure 3. The results after calculating Pointfuse

3. RESULTS AND EVALUATION

In this study, dense point clouds in urban area are obtained by software SURE developed by nframes company. The 3D vectors of roofs are extracted by software Pointfuse.

3.1 Test Data

The aerial images used for calibration purpose flown over the Nangang camera calibration field located in Nantou county, Taiwan served as our test data. The calibration field of Nangang, as Figure 4 shows, contain a large number of buildings and many kinds of building roof types. Thirty aerial images taken by UltraCam Xp-Wide-Angle in 2012 are used in this study. The image size is 11,310 rows times 17,310 columns with pixel size 6 micron and focal length 70.5 mm. The image set is captured with 80% end-lap and 60% side-lap from average flight height of 540 m, and the ground sampling distance (GSD) is 4.6 cm. Each image has ground coverage of 796 m x 520 m. Such high overlapping images can provide sufficient redundancy and increase the chance to obtain some usable results in areas of break lines, shadow and occlusion during the dense matching process.

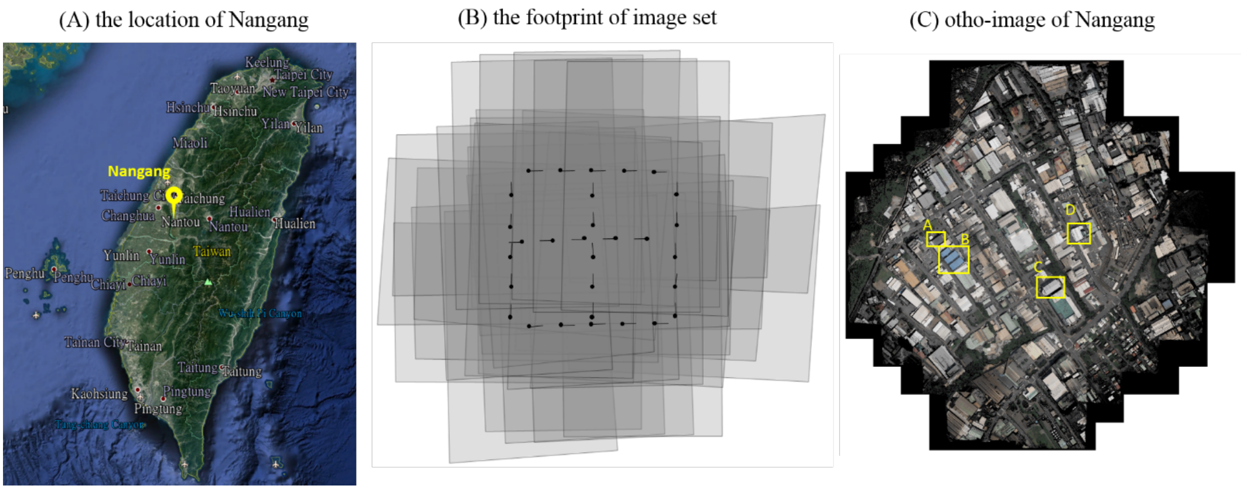
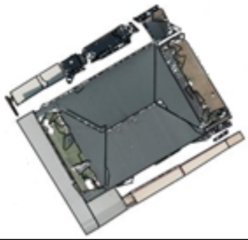
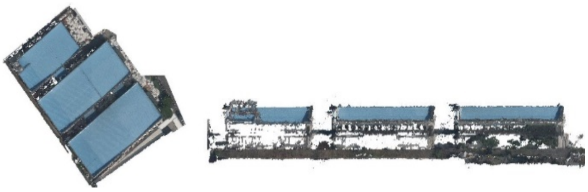
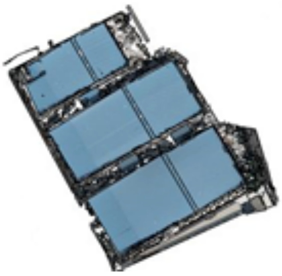


Figure 4. The basic information of test imagery

3.2 Results of Dense Matching and Reconstructing 3D Building Models

In this test area, we obtain 2,671,825,124 3D points from 30 images with SURE software. The imagery covers an area of 1.62 km^2. Therefore, the point density is 1649.27 points/m^2. In order to assess the effects of the dense matching results and the influence of complexity of roof types on the extracting of roof edges, four sections are selected as shown in Table 1 . Case A is a typical hipped roof; Case B are three adjacent houses with narrow gaps in between where blunders might occur from dense matching; Case C contains two different kinds of building shapes. One is semicircular building and the other is roof ridge building. Case D is a more complex house of which roof includes a ridge roof and a flat cover on its top. The location of those four cases are labelled at Figure 4(C).

Table 1. The results of dense matching and 3D models reconstruction of four test areas

	Aerial image	Dense point clouds	3D vectors of roof
A			
B			
C			
D			

3.3 Difficulty in Reconstructing 3D Building Models

The results of dense matching and the subsequent vector extraction are shown in the second and third column in Table 1. At first, we can find that there are fewer point clouds on side walls due to occlusion. For example, in case A and C, the 3D points located on side walls are sparse because the distance of two building is very close so that only very small part of the walls could be seen from aerial images. In this situation, a LoD2 house which should contain building installations attached to the side walls is already very difficult to be reconstructed, not to mention a LoD3 building for which doors and windows should also be reconstructed. The areas with break lines are prone to having blunders, like Figure 5(A) shows. Blunders cause wrong reconstruction of vectors, like Figure 5(B) shows. This shows unless the dense matching algorithm has a very efficient blunder detection mechanism, the reconstructed vectors cannot be error-free and need to be corrected manually in order to get correct 3D models.

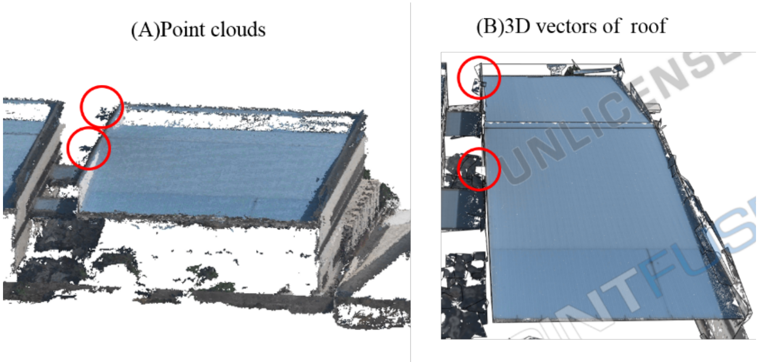


Figure 5. The results of 3D vectors of roof by using dense point clouds containing blunders

As to the 3D vectors of roofs, for case A, B and C vectors depicting the roofs have been correctly reconstructed almost entirely. However, in case C, the arc of the semicircular building is resolved in a large number of piecewise linear vectors instead of a single arc, as Figure 6(A) shows, and that doesn't comply with the CityGML standard. We also choose another arched roof to do the same test, and the result as Figure 6(B) shows, it is reconstructed as a gabled roof of two flat planes instead of an arch. In Taiwan, there are various shapes of roof which can't be described in simple straight lines. To improve the quality of reconstruction, we suggest that it is a necessity that curved lines could also be included in the automatic vector extraction algorithms. In case D, as mentioned before, which shows a more complex roof structure, we can find that 3D vectors of roof have not been extracted completely, as Table 1 shows. The roof reconstructed is composed of too many polygons which don't comply with the CityGML. To demonstrate how that roof surface of LoD2 should be, we manually reconstructed the 3D vectors of the roof, like Figure 7 shows, and the complete building top should include a ridge roof and a flat cover on the ridge. There are many other more complex additional structures on top of the buildings, such as water tanks, pigeon houses and so on are commonly seen in Taiwan. They all increase the difficulty in reconstructing of LoD2 models.

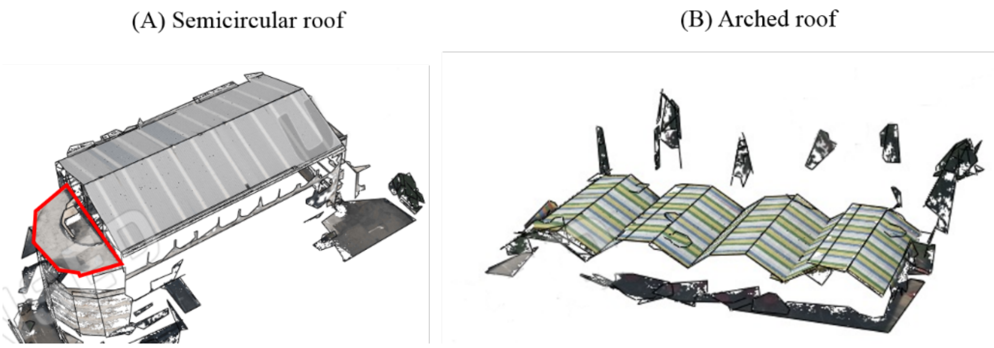


Figure 6. The results of 3D vectors of roof of semicircular roof and arched roof

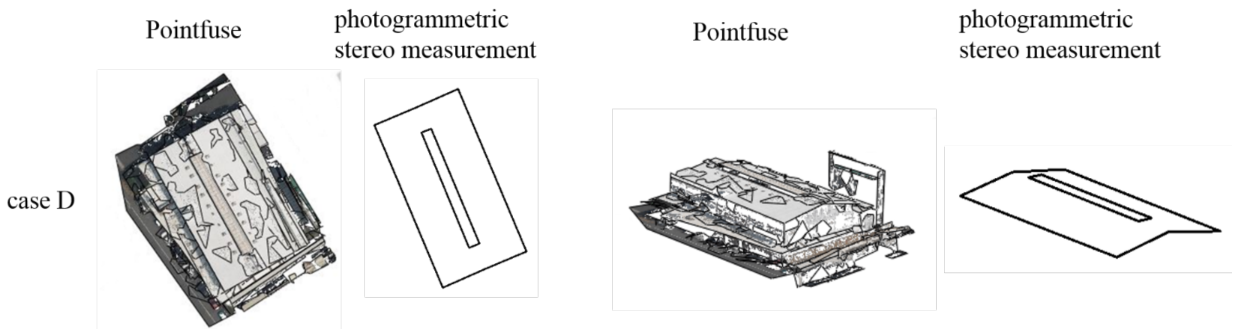


Figure 7. The 3D roof vectors extracted by Pointfuse and photogrammetric stereo measurement

Moreover, there are many town houses in Taiwan which share the same roof and thus can't be distinguished in aerial images from above, like Figure 8 shows. Point clouds of such town houses does not provide any information about how many separate houses under the roof. Automatic vector reconstruction for 3D models fails here. Because under such roofs only one building will be reconstructed. The standard of CityGML demands that the roof should be individually reconstructed and attributes be assigned for each house respectively. If we want to solve this problem, we have to utilize the extra information, for example, using cadasters to divide automatically reconstructed 3D vectors of roof into separate buildings.

Figure 8. Town house in Taiwan(source: Google Maps)

4. CONCLUSIONS

This study analyzes the difficulty in generating 3D city models according to the CityGML standard from the results of dense image matching alone. Our results indicate that the blunder removal in dense point clouds, the algorithms used to extract 3D vectors of roofs, the complexity of roof types, etc., would affect the quality of the reconstructed 3D city models. Currently fully automatic generation of 3D building models from densely matched point clouds according to CityGML standard is still impossible. Manual modification of the automatically reconstructed vectors is necessary.

5. REFERENCES

Bleyer, M., Rhemann, C., and Rother, C., 2011. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In Jesse Hoey, Stephen McKenna and Emanuele Trucco, Proceedings of the British Machine Vision Conference, pp. 14.1-14.11.

Brunn, A., and Weidner, U., 1997. Extracting Buildings from Digital Surface Models, ISPRS Archives, Stuttgart, Vol. 32, Part 3-4W2, pp.27-34.

Fromherz, T., and Bichsel. M., 1995. Shape from multiple cues: In-tegrating local brightness information. In ICYCS.

Fuchs, C., Gülch, E. and Förstner, W., 1998. OEEPE Survey on 3D-City Models, OEEPE Official Publication #35.

Furukawa, Y., and Ponce, J., 2010. Accurate, Dense, and Robust Multiview Stereopsis. IEEE Transactions On Pattern Analysis and Machine Intelligence. Vol. 32, No. 8, pp. 1362-1376.

Haala, N., and Brenner, C., 1999. Virtual City Models from Laser Altimeter and 2D Map Data, Photogrammetric Engineering & Remote Sensing, Vol. 65, pp. 787-797.

Hirschmüller, H., 2005. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 807-814.

Kutulakos K., and Seitz, S., 2000. A theory of shape by space carving. IJCV, 38(3), pp.199-218.

OGC, 2008. OpenGIS City Geographic Markup Language (CityGML) Encoding Standard.

OGC, 2012. OGC City Geographic Markup Language (CityGML) Encoding Standard.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. In Proceedings of the IEEE conference on computer vision and pattern recognition, Vol. 1, pp. 519-526.

Schnabel, R., Wahl, R., and Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. Computer Graphics Forum 26(2), pp. 214–226.