EXTRACTION OF BUILDINGS
FROM HIGH-RESOLUTION SATELLITE DATA AND LIDAR

G. Sohn

Dept. of Geomatic Engineering, University College London, Gower Street, London, WC1E 6BT UK -
gsohn@ge.ucl.ac.uk

Commission WG III/4

KEY WORDS: IKONOS, LIDAR, Building Reconstruction

ABSTRACT:

Acquiring accurate detection and description of buildings is a difficult object recognition problem due to a high complexity of the scene content and the object representation. Since most urban scenes deliver very rich information, a robust separation of fore (objects to be reconstructed) from background (irrelevant features) is an essential process in object recognition system, but is difficult to achieve since objects in the scenes normally show a large amount of geometric or chromatic co-similarity across them. In addition, most mapping applications require building shapes to be reconstructed with a high degree of geometric freedom. However, information extracted from remotely sensed data is usually incomplete for reconstructing a full description of building objects due to limited resolving power of the sensor used, object complexity, disadvantageous illumination condition and perspective views. The research illustrates that the problems outlined above can be resolved by fusing multi-data sources, where 2D linear features extracted from Ikonos images is attributed with a high-quality of 3D information provided by airborne lidar.

1. INTRODUCTION

With the recent advent of a series of commercialized high-resolution satellite, the potential of Ikonos imagery in topographic mapping has been investigated and highlighted by many researchers (Holland et al., 2002; Holland & Marshall, 2003). However, the success of fully automated reconstruction of building objects from the Ikonos imagery is still far to reach, and only partial solution in constrained environments have been reported (Kim & Muller, 2002; Lee et al, 2003). This research aims to develop a building extraction system which automatically reconstructs prismatic building models in an urban environment. In particular, two research interests have been exploited in this study; *building detection* (separation of objects to be reconstructed from irrelevant features) and *building description* (reconstruction of generic shape of building boundaries in a combination of data-driven and model-drive cues).

2. DATA CHARACTERISTICS

2.1 Ikonos image

A "pan-sharpened" multi-spectral (PSM) Ikonos image covering the Greenwich industrial area was provided by Infoterra Co. for this research. The Ikonos PSM image is produced by combing the multi-spectral data with the panchromatic data, and resampled with 1-metre ground pixel. The image was orthorectified by Space Imaging Co. to satisfy the positional accuracy (~1.9 metres) of Precision Product of Space Imaging. Figure 1 shows the Greenwich Ikonos PSM image, in which the red channel is replaced with the near-infrared channel while the green channel as red channel respectively. The sub-scene image is 681 x 502 pixels with 1m resolution whose dimension of the image is approximately 341,862 m^2.

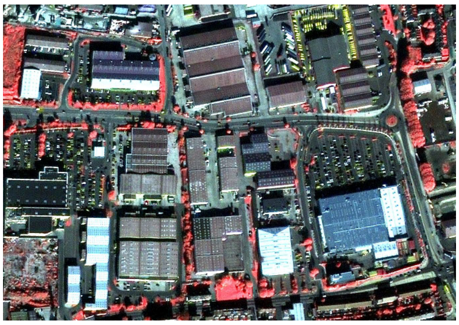


Figure 1. Greenwich Ikonos PSM image

2.2 Lidar DSM

Figure 2 shows a lidar DSM which was also provided by Infoterra Co., which covers a sub-site of Greenwich industrial area with the size of 305,523 m^2. The lidar DSM was acquired by the first pulse of OPTEC 1020 airborne laser sensor. The data has been converted from OSGB36 (plan) and OSD Newlyn (height) to UTM/WGS84. The lidar DSM contains a total of 30,782 points, which corresponds to a point density of 0.1 (points/m^2), i.e., one point per 3.2 x 3.2 (m^2). The height of the study area varies from 1.4 m to 26.3 m. The terrain in the Northern part is higher than the Southern part, and the highest terrain height can be found in the North-West corner in figure 2. The Greenwich LIDAR DSM shows a typical urban environment, where a number of industrial buildings with different sizes spread over the study area. In particular, figure 2 shows the point density of the OPTEC 1020 LIDAR is not enough to properly represent the shape of those small houses though they are formed in planar roof surfaces.

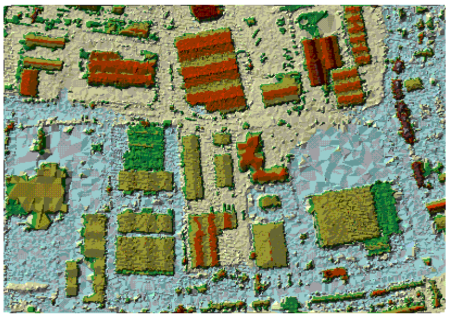


Figure 2. Greenwich lidar DSM

3. BUILDING DETECTION

The complexity of building extraction process can be reduced by a large amount if the process can be focused on single building object. This section presents a building detection method to localize individual buildings by sequentially removing dominant urban features which are not relevant to buildings.

3.1 Terrain detection

A lidar filter, called recursive terrain fragmentation (RTF) filter, was developed to distinguish between on-terrain points and off-terrain ones from a cloud of lidar points. The RTF filter was implemented by employing a hypothesis-test optimization in different scales. This filter assumes that generic terrain surface is a mosaic of planar terrain surfaces. The entire lidar space, initially hypothesized as a single planar terrain surface, is recursively fragmented with small sub-regions until the coexistence of different terrain slopes cannot be found over all fragmented regions. More detailed description of the RTF filter can be found in Sohn & Dowman (2002). Figure 3(a) shows the on-terrain points detected by the RTF filter from figure 2. In this figure, some terrain segments which are not densely covered by the filtered on-terrain points show poor quality of the Greenwich lidar DSM.

3.2 High-rise and low-rise object detection

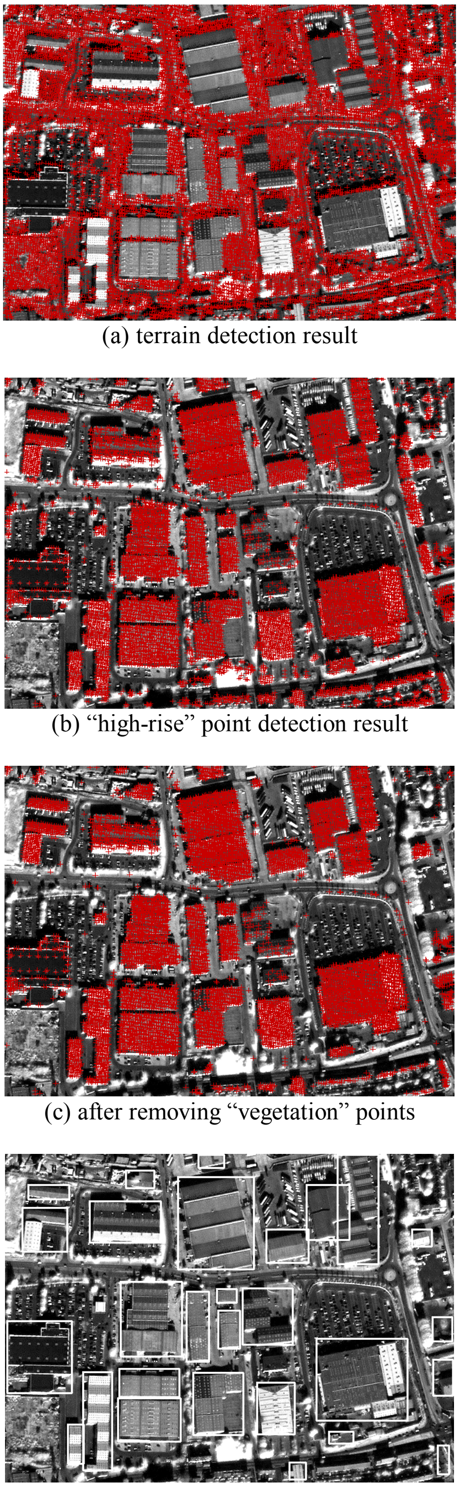
With the on-terrain points detected by the RTF filter, a DTM is generated. Then, outlying points with a height less than a pre-defined height threshold (4m) from the generated DTM are classified as "high-rise" features, otherwise as the "low-rise" ones (see figure 3(b)).

3.3 Tree detection

Since the "high-rise" feature class generally contains trees as well as buildings, "vegetation" points must be identified. The normalized difference vegetation indices (NDVI) are computed by a combination of red and near-infrared channels of Ikonos. When the "high-rise points" are back-projected onto the NDVI map, a small mask (5x5 size) is constructed around them, and "vegetation" points are identified if any masked pixel has the NDVI value larger than a threshold value (>0.8) (see figure 3(c)).

3.4 Building blob detection

Isolating the building label points and making them into individual building objects is rather straightforward. Those points classified into the on-terrain, low-rise, and tree objects are together assigned non-building labels. Then, building points surrounded by the non-building labels, are grouped as isolated objects. As a result, 28 building "blobs" can be found from figure 3(d) after removing small "blobs" whose member points are less than 30 points. Further processing allows the individual building "blobs" to be bounded with rectangle polygons, and these polygons are then fed into the building description process, which will be discussed in the next section.



(a) terrain detection result

(b) "high-rise" point detection result

(c) after removing "vegetation" points

(d) building "blob" detection result

Figure 3. Building detection results

4. BUILDING DESCRIPTION

This section presents a building description process which reconstructs building outlines from each building "blob". Generic building shape is represented as a mosaic of convex polygon. A set of linear cues is extracted by both data-driven and model-driven approaches. The building "blobs" are recursively intersected by those linear cues, which produces a set of polygon cues. Finally, building outlines are reconstructed by merging only "building" polygons forming building objects.

4.1 Data-driven linear cue extraction

The first stage of the building description is to extract boundary lines from Ikonos imagery with the support of the RTF filtering result. Straight lines extracted by the Burns algorithm (Burns et al., 1986) are filtered by a length criterion, by which only lines larger than pre-specified length threshold, l_d =5m, remain for further processing. Then, two rectangle boxes with certain width, l_w=5m, are generated along two orthogonal directions to the line vector filtered in length. The determination of boundary line can be given if non-building and building points are simultaneously found in both boxes or if only building-label points are found in one of the boxes and no lidar point can be found in the other box. The latter boundary line condition is considered if a low density lidar dataset is used. Figure 4 illustrates this.

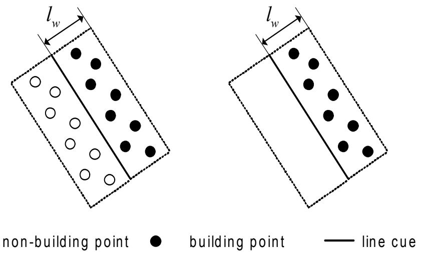
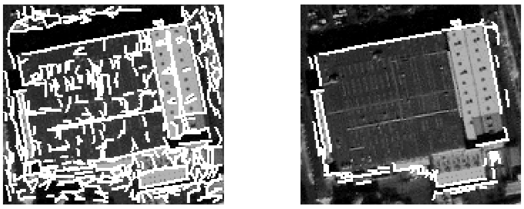


Figure 4. Illustration of boundary line detection



(a) extracted straight lines (b) filtered boundary lines

Figure 5. Result of data-drive cue extraction

As a final line filtering process, a geometric disturbance corrupted by noise is regularized over boundary lines. A set of dominant line angles of boundary lines is analyzed from a gradient-weighted histogram which is quantized in 255 discrete angular units. In order to separate a weak, but significant peak from other nearby dominant angles, a hierarchical histogram-clustering method is applied. Once the dominant angle, θ_d, is obtained, lines with angle discrepancies which are less than certain angel thresholds, θ_{th}=30°, from θ_d are found. Then, their line geometries are modified as their angles are replaced with θ_d. These modified lines do not contribute to the succeeding dominant angle analysis and the next dominant angle is obtained. In this way, a set of dominant angles is obtained, by which geometric properties of boundary lines can be regularized (see figure 5).

4.2 Model-driven linear cue extraction

New line cues are "virtually" extracted from lidar space in order to compensate for the lack of intensity line cue density by employing specific building models. For each intensity line cue, parallel lines and "U" structured lines are inferred from lidar space. First, a box growing direction, pointing to the location of parallel boundary line is determined. To this end, a small virtual box is generated with a width of l_w=5m from the selected intensity line in the same way of detecting boundary lines presented in §4.1. To that direction, the virtual box grows until it comes across any on-terrain point (see figure 6 (a)). Then, it de-grows in order to have maximum building points while in its minimum size (see figure 6 (b)). In this way, the virtual box is expanded, but at this time, towards to two orthogonal directions to the parallel boundary line detected (see figure 6 (c)). Thus, "U" structured boundary lines made with the parallel boundary line can be detected. Finally, these three virtual lines detected are back-projected onto image space and then, their line geometry is adjusted by gradient weighted least-square method. Figure 6(d) shows model-driven cues extracted from figure 5(b).

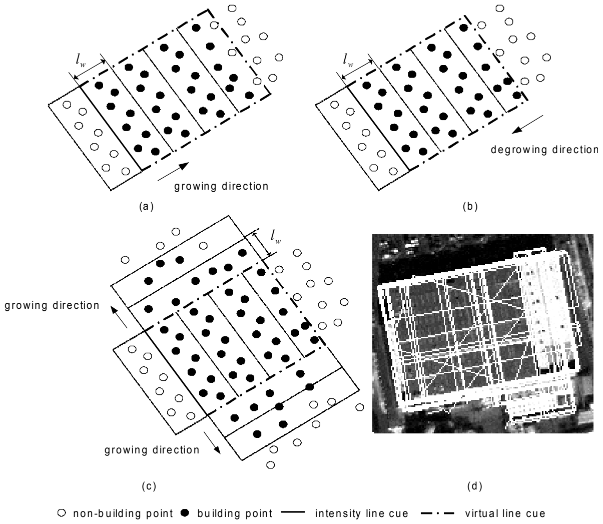


Figure 6. Result of model-driven cue extraction

4.3 Polygonal cue generation

Initial polygons resulting from the building detection result of figure 3(d) are decomposed of a set of convex polygons by a recursive intersection of linear cues, called hyperlines. This polygonal segmentation is implemented by BSP (Binary Space Partitioning) tree algorithm introduced by Fuchs et al. (1980). Figure 7 illustrates the overall partitioning scheme to generate polygons. Suppose that we have an initial polygon with rectangle geometry, P^0, wherein LIDAR points are distributed with building and non-building label. All vertices comprising P^0 are stored as a root node of BSP tree for further recursive partitioning (see figure 7(a)).

A set of hyperlines, $\{h^i{:}i{=}1,\ldots,\mathrm{N}\}$, are computed as P^0 is intersected respectively by a set of line segments, $\{l^i{:}\mathrm{i}{=}1,\ldots,\mathrm{N}\}$, which is constructed by integrating the data-driven and model-driven line cues. After setting up the hyperline list, a process to partition P^0 with hyperlines starts. This partitioning process consists of two procedure; *polygon classification* and *partition scoring*.

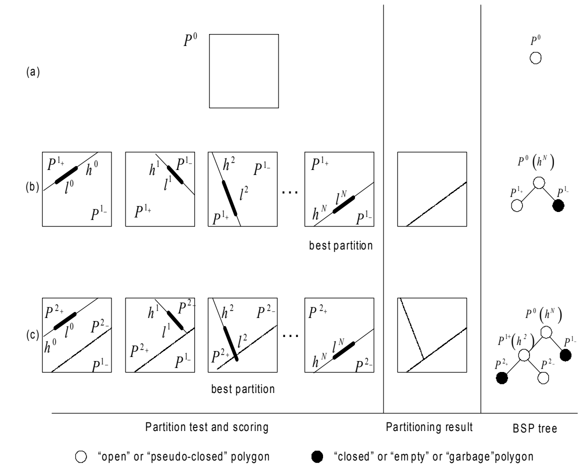


Figure 7. Illustration of polygonal cue generation

4.3.1 Polygon classification

Polygon classification is a process to determine whether or not the partitioning process is triggered over a given polygon, P^i. A polygon, P^i, is classified into a number of polygon classes; "empty", "open", "closed", "pseudo-closed", "garbage" polygon. These polygon classes are pre-determined depending on the labelling attributes of the member points of P^i or point density of the member points of P^i or geometric property of P^i as follows:

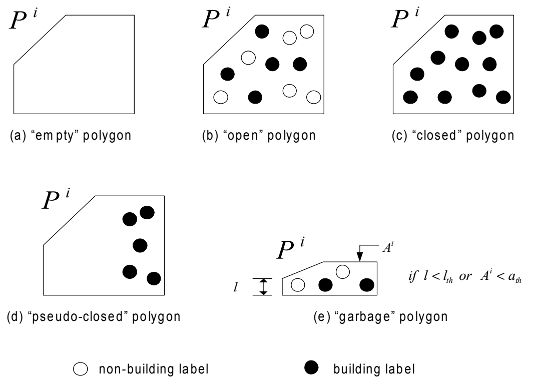


Figure 8. Polygon classification

- ***"Empty" polygon***: P^i is classified as "empty" polygon if there is no member point within P^i (see figure 8 (a)).
- ***"Open" polygon***: P^i is classified as "open" polygon if the member points of P^i are attributed with both building and non-building labels (see figure 8 (b)).
- ***"Closed" polygon***: P^i is classified as "closed" polygon if the member points of P^i are attributed with only building label (see figure 8 (c)).
- ***"Pseudo-closed" polygon***: P^i is classified as "pseudo-closed" polygon if the member points of P^i are attributed with only building label, and the point density of P^i, $d_{pt}(P^i)$, is less than d_{th}=0.1 (see figure 8 (d)), where $d_{pt}(P^i)$ is determined by

$$d_{pt} = \frac{N_{mem}\left(P^i\right)}{A^i} \tag{1}$$

  where $N_{mem}(P^i)$ and A^i are the number of member points and the area of P^i respectively.
- "Garbage" polygon: P^i is classified as "garbage" polygon if the member points of P^i are attributed with both building and non-building labels, and any lateral length or the area P^i is less than a certain threshold, i.e., l_{th}= 5 and a_{th}=50 respectively (see figure 8 (e)).

The P^i is partitioned with two child convex polygons if it is classified as "open" or "pseudo-closed" polygon; otherwise the partitioning over P^i is terminated.

4.3.2 Polygon scoring

Once the partitioning of P^i is determined through the polygon classification. The second step is *partition scoring*. This process determines a hyperline, h^*, to generate the "best" partitioning result of P^i from the hyperline list. The selection of h^* is achieved by a partition scoring function. That is, all the hyperlines are tested to obtain the "best" partition of P^i and the partitioning result generated by each hyperline is evaluated by a partition scoring function. A hyperline, h^*, with the highest partitioning score is finally selected to partition P^i. The partition scoring function, H, over a polygon, P^0, is given by

$$H\left(P^0;h^i,l^i\right) = \arg\max\left(H\left(P^{1+};h^i,l^i\right),H\left(P^{1-};h^i,l^i\right)\right) \tag{2}$$

where P^{1+} and P^{1-} are child polygons produced by halving P^0 with a hyperline, h^i. In Eq 2, H assigns a maximum score to h^i if it produces the best partitioning result, whereas a minimum score for the worst partitioning result. Also, H differently computes scores depending on the polygon class of P^0.

If P^0 is classified as the "open" polygon, H computes partitioning scores according to a bias degree of label distribution over P^{1+} and P^{1-} divided by h^i; H for "open" polygon computes higher partitioning score when a "closed" polygon with larger area is produced by h^i (see figure 9 (a)). The partition scoring function, H, for "open" polygon can be described by

$$\begin{aligned} H\left(P^{1+};h^i,l^i\right) &= \frac{1}{2}\left\{\frac{N_{bld}\left(P^{1+};h^i,l^i\right)}{N_{bld}\left(P^0;h^i,l^i\right)} + \frac{N_{non-bld}\left(P^{1-};h^i,l^i\right)}{N_{non-bld}\left(P^0;h^i,l^i\right)}\right\} \\ H\left(P^{1-};h^i,l^i\right) &= \frac{1}{2}\left\{\frac{N_{bld}\left(P^{1-};h^i,l^i\right)}{N_{bld}\left(P^0;h^i,l^i\right)} + \frac{N_{non-bld}\left(P^{1+};h^i,l^i\right)}{N_{non-bld}\left(P^0;h^i,l^i\right)}\right\} \end{aligned} \tag{3}$$

where $N_{non\text{-}bld}$ and N_{bld} are functions to count numbers of building labels and non-building labels belonging to a corresponding polygon.

If P^0 is classified as the "pseudo-closed" polygon, H computes the partitioning score by an area ratio of child "empty" polygon over P^0 when either of P^{1+} and P^{1-} is recognized as an "empty" polygon, otherwise the null value is assigned as the partitioning score for P^0 by H. The "best" partition of P^0 can be obtained when an "empty" polygon with the largest area is produced by h^i (see figure 9 (b)). The partition scoring function, H, for "pseudo-closed" polygon can be described by

$$\begin{cases} H\left(P^{1+};h^i,l^i\right)=\dfrac{A\left(P^{1+}\right)}{A\left(P^o\right)} & \text{if } P^{1+}=\text{"empty" polygon} \\ H\left(P^{1-};h^i,l^i\right)=\dfrac{A\left(P^{1-}\right)}{A\left(P^o\right)} & \text{if } P^{1-}=\text{"empty" polygon} \end{cases} \qquad (4)$$

$$\begin{cases} H\left(P^{1+};h^i,l^i\right)=0 & \text{if } P^{1+}\neq\text{"empty" polygon} \\ H\left(P^{1-};h^i,l^i\right)=0 & \text{if } P^{1-}\neq\text{"empty" polygon} \end{cases}$$

where $A()$ is the area of corresponding polygon. In fact, the partition functions defined in Eq. 3 and Eq. 4 generate polygons according to their level-of-detail forming a building object; the most "significant" building part is generated first and less "significant" one is later.

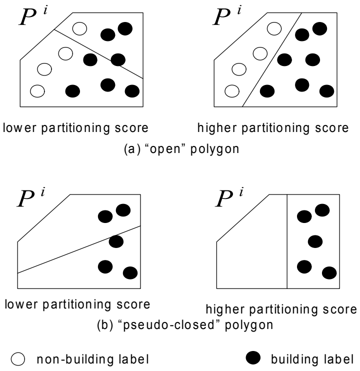


Figure 9. Illustration of partition scoring functions

Once the partitioning scores for h^i are computed by Eq. 3 or Eq. 4, remaining hyperlines are sequentially selected from the hyperline list and their partitioning scores are measured by H. In figure 7(b), a hyperline, h^N, with the maximum partitioning score is finally selected to partition P^0. Then, geometric information of P^0 and h^N are stored as a root node of BSP tree, which is expanded as new child nodes with vertices of P^{1+} and P^{1-} are added to the root node for further recursive partitioning. The same method used for the partition of P^0 is applied to P^{1+} and P^{1-} respectively, but to only an "open" or "pseudo-closed" polygon. This process continues until no leaf node of the BSP tree can be partitioned by hyperlines (see figure 7 (c)).

4.4 Polygonal cue grouping

Figure 10 (a) – (c) shows an example how the BUS space with a set of convex polygons is generated by the recursive partition of an initial polygon as described in the previous section.

Once the BUS space is generated by expanding a BSP tree, final leaves of the BSP tree are collected. A heuristic filtering is applied to them so that only "building" polygons remain (see figure 10(d)). A convex polygon of final leaves of the BSP tree is verified as the "building" polygon by following rules:

- A polygon, P^i, is verified as the "building" polygon If it is classified as "closed" polygon, satisfying following conditions:

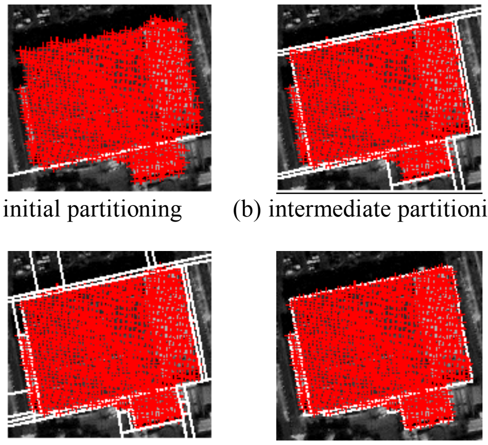
$$N_{mem}\left(P^i\right)>n_{th} \text{ and } d_{pt}\left(P^i\right)>\gamma\times d_{th};\ P^i=\text{"closed" polygon} \quad (5)$$

where N_{mem} is the number of member points of P^i; n_{th} (=5) is a member point threshold; d_{pt} is the point density of P^i computed by Eq. 1; γ (=0.6) is a control parameter ($0\leq\gamma\leq1$); d_{th} (=0.1) is a point density threshold.

- A polygon, P^i, is verified as the "building" polygon if it is classified as "open" polygon, satisfying following conditions:

$$\rho_{pt}\left(P^i\right)=\frac{N_{bld}\left(P^i\right)}{N_{mem}\left(P^i\right)}>\rho_{th};\ P^i=\text{"open" polygon} \qquad (6)$$

where ρ_{pt} is a point ratio of building labels over the total number of member points of P^i and its threshold is ρ_{th} (=0.6); N_{bld} and N_{mem} are functions to count numbers of building labels and non-building labels belonging to P^i.



(a) initial partitioning (b) intermediate partitioning

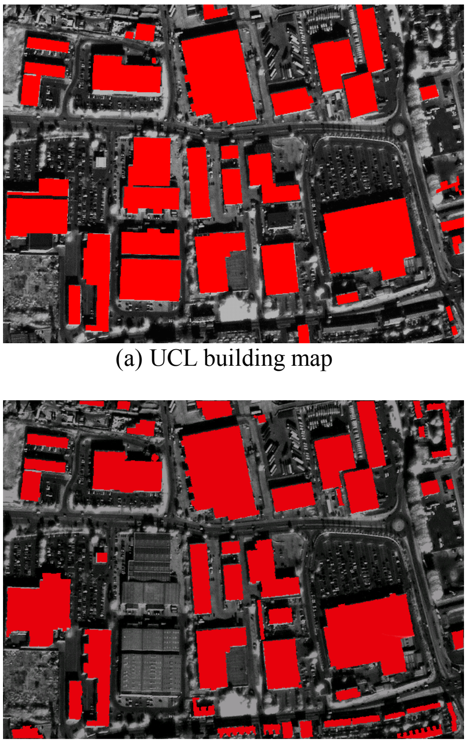
(c) final BSP partitioning (d) "open" polygon filtering

Figure 10. Polygonal cue generation and grouping

5. BUILDING EXTRACTION RESULT

Figure 11(a) shows a building extraction result over the Greenwich dataset (referred as the UCL building map) generated by the proposed technique. The overall success of the technique was evaluated in comparison with the ground plan vectors of MasterMap® provided by the Ordnance Survey (see figure 11(b)).

Although the OS MasterMap® provides a high-level of detail and accuracy, there are distracting features of the OS MasterMap® that causes difficulties in the quality assessments. The OS data does not contain some buildings even though they are obviously apparent in the Ikonos image and lidar. This is because the OS data was constructed at a different time to the acquisition of the Ikonos image and lidar data, from which the UCL building map was generated. In addition, the scale of the features in the OS MasterMap® has been compiled at a larger scale than the one of Ikonos image. As a result, very small features cannot be clearly recognized in the Ikonos image. Since this research has an interest to evaluate the quality of the UCL building map, rather thane the OS MasterMap®, those inherent faults of the OS MasterMap® was removed from the UCL data, and small polygons whose member points are less than 100 points were also excluded before the quality evaluation.



(a) UCL building map

(b) OS MasterMap® ground plan

Figure 11. Building extraction result and OS MasterMap®

A number of objective evaluation metrics suggested by Shufelt (1999) was adopted in order to provide a quantitative assessment of the developed building extraction algorithm. These metrics can be defined as follows:

$$\begin{aligned} &\textit{building detection percentage}\ (\%) = 100 \times TP/(TP+TN) \\ &\textit{branching factor} = FP/TP \qquad (8) \\ &\textit{quality percentage}\ (\%) = 100 \times TP/(TP+FP+FN) \end{aligned}$$

where *TP* (True Positive) is a building classified by both datasets, *TN* (True Negative) is a non-building object classified by both datasets, *FP* (False Positive) is a building classified only by the UCL building map, and *FN* (False Negative) is a building classified only by the OS MasterMap®. Table 1 shows the pixel classification results, and the evaluation on the UCL building map computed by Eq. 8 is presented in table 2.

Table 1. Pixel classification results

Pixel classification	Pixels
True Positive (TP)	67085
True Negative (TN)	255794
False Positive (FP)	4344
False Negative (FN)	14639

Table 2. Building extraction metric result

Building extraction metric	Evaluation result
Building detection percentage	93.92 (%)
Branching factor	0.22
Quality percentage	77.94 (%)

6. DISCUSSION

As can be seen in table 2, the proposed building extraction technique detected building objects with 94 % detection rate (*building detection percentage*), and showed 0.2 delineation performance (*branching factor*). Finally, the overall success of the technique was evaluated as 78 % extraction quality (*quality percentage*). These results suggest that the developed system can successfully acquire accurate detection and description of building objects using Ikonos images and lidar data with a moderate point density.

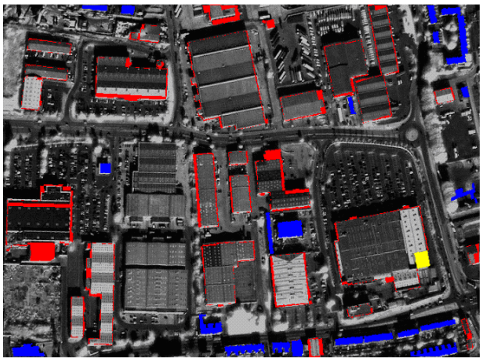


Figure 12. Building extraction errors

However, the UCL building map contains certain amount of building extraction errors (*FP* and *FN*), which should be reduced for achieving a more accurate extraction of building objects. The errors apparent in the result generated by the developed system can generally be divided into three categories:

Building detection error: most of *FN* pixels in Eq. 8 were generated by under-detection of the terraced houses (see blue coloured polygons in figure 12). This problem is mainly caused by the fact that the NDVI classification described in §3.3 tends to over-remove "building" points over those building with long and narrow structures such as a row of terraced houses and results in a very small "blob", whose member points are fewer than 30 points. This problem can be resolved by modifying the NDVI classification from point-wise to region-wise approach. That is, in order to ensure larger numbers of member points are obtained, "high-rise" points populated in §3.2 are clustered in a number of single objects, and then a cluster-by-cluster tree detection is made by the NDVI classification. This modification may make terraced houses detectable since more member points are retained.

Building delineation error: these errors are caused when boundaries of building objects are not properly extracted by the building description process (see red coloured pixels in figure 12). Those errors are related to the inherent planimetric accuracy of input data (i.e., Ikonos image, lidar data, and OS MasterMap®), and the point density of lidar data. Most of boundary delineation errors are deviated from the OS reference data with one or two pixels if lidar measurements are sufficiently acquired over buildings (see figure 13 (a)). However, as lidar points are acquired with less point density over a building, more errors are produced around its boundaries (figure 13 (b) and (c)). This is because the detection of data-driven lines and model-driven lines is more difficult over a building with coarser point density than the one with denser point density. As a result, mis-location of data-driven lines and model-driven lines leads to the generation of delineation errors around building boundaries.

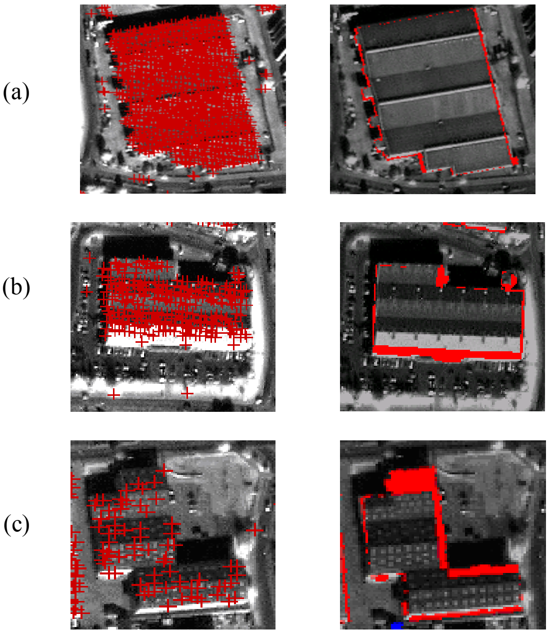


Figure 13. Building delineation errors; the first column shows cut-out Ikonos images overlaid with building-labelled lidar points; the second column shows building delineation errors

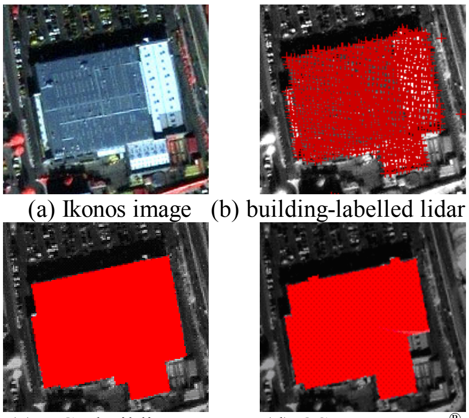


Figure 14. Reference data errors

Reference data error: these errors are caused by the inherent faults in the OS MasterMap® (see yellow coloured pixel in figure 12). As can be seen in figure 14, the UCL building map can successfully delineate boundaries of a building based on the result of lidar measurements and Ikonos image. However, the OS building map missed some part of that building (cf. figure 14 (c) and (d)). As outlined earlier, this error caused by a time difference between the acquisition of the Ikonos image and lidar data, and the construction of the OS data. The analysis of the reference errors suggests that the developed building extraction technique can be also used for applications detecting changes in an urban environment and supporting map compilation.

7. CONCLUSIONS

This paper presented a system for automatically detecting building objects and delineating their boundaries from Ikonos images and lidar data. A few new ideas to combine complementary nature of intensity images and high-quality of 3D information to solve problems associated with building detection and building description are introduced. The overall success of the developed building extraction system was evaluated in comparison with the OS MasterMap® ground plan. The results highlights Ikonos images can be used in topographic mapping at large scale in a combination of lidar data. The current system is limited to delineating polygonal shapes of buildings with flat roofs. Thus, a further development must be directed to reconstruct 3D roof structures based on the ground plans extracted by the current techniques.

REFERENCES

Burns, J.B., Hanson, A.R. and Riseman, E.M., 1986. Extracting straight lines. *IEEE Pattern Analysis and Machine Intelligence,* 8(4):425-455.

Fuchs, H., Kedem, Z.M. and Naylor, B.F., 1980. On visible surface generation by a priori tree structures. *Computer Graphics*, 14(3):124-133.

Holland, D., Guilford, B. and Murray, K., (eds) 2002. OEEPE-Project on topographic mapping from high resolution space sensors. Media-Print, Paderborn.

Holland, D. and Marshall, P., 2003. Using high-resolution satellite imagery in a well-mapped county. *Proceedings of Joint ISPRS Workshop on High Resolution Mapping from Space 2003*, Hannover, Germany, 6-8 October 2003. CD-ROM.

Kim, J.R. and Muller, J-.P., 2002. 3D reconstruction from very high resolution satellite stereo and its application to object identification. *Proceedings of ISPRS technical Commission IV Symposium 2002 on Geospatial theory, Processing and Applications*, Ottawa, Canada, July 9-12, 2002, 34(4). CD-ROM.

Lee, D.S., Shan, J. and Bethel, J.S., 2003. Class-guided building extraction from Ikonos imagery. *Photogrammetric Engineering & Remote Sensing*, 69(2):143-150.

Shufelt, J.A., 1999. Performance evaluation and analysis of monocular building extraction from aerial imagery. *IEEE Pattern Analysis and Machine Intelligence*, 21(4):311-326.

Sohn, G. and Dowman, I.J., 2002. Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. In: Proceedings the Photogrammetric Computer Vision, Graz, Austria, September, pp.336-344.