AUTOMATIC DETECTION AND FEATURE ESTIMATION OF WINDOWS FOR REFINING BUILDING FAÇADES IN 3D URBAN POINT CLOUDS

A. K. Aijazi[a,b,*], P. Checchin[a,b] and L. Trassoudaine[a,b]

[a] INSTITUT PASCAL, Université Blaise Pascal, Clermont Université, BP 10448, F-63000 Clermont-Ferrand, France
[b] INSTITUT PASCAL, CNRS, UMR 6602, F-63171 Aubière, France
kamalaijazi@gmail.com; paul.checchin@univ-bpclermont.fr; laurent.trassoudaine@univ-bpclermont.fr

Commission III

KEY WORDS: 3D LiDAR point clouds, Automatic window detection, 3D urban cartography

ABSTRACT:

This paper presents a method that automatically detects windows of different shapes, in 3D LiDAR point clouds obtained from mobile terrestrial data acquisition systems in the urban environment. The proposed method first segments out 3D points belonging to the building façade from the 3D urban point cloud and then projects them onto a 2D plane parallel to the building façade. After point inversion within a watertight boundary, windows are segmented out based on geometrical information. The window features/parameters are then estimated exploiting both symmetrically corresponding windows in the façade as well as temporally corresponding windows in successive passages, based on analysis of variance measurements. This unique fusion of information not only accommodates for lack of symmetry but also helps complete missing features due to occlusions. The estimated windows are then used to refine the 3D point cloud of the building façade. The results, evaluated on real data using different standard evaluation metrics, demonstrate the efficacy as well as the technical prowess of the method.

1 INTRODUCTION

Analysis of building façades for 3D building reconstruction and realistic geometrical modeling of the urban environment has lately received considerable attention. Such realistic models play an important role in several applications pertaining to different perception tasks such as driving assistance, autonomous navigation through the urban environment, fly-through rendering, virtual tourism and urban mission planning, etc. Driven by an increasing demand to improve the quality of such applications, some work has been done lately, focusing on the semantic analysis of building façades including detecting and modeling geometric structures like windows and doors.

Usually, such applications rely on data acquired from mobile terrestrial data acquisition systems. These ground-based or vehicle-borne laser scanning techniques help acquire highly accurate georeferenced 3D points with sufficient details due to their close-range data collection. However, this task is made significantly harder by complex representations in such a highly cluttered urban environment (cars, poles, trees, vegetation, etc.), variability of appearances (e.g. open/closed windows and doors, etc.) and missing data due to physical properties. The problem is further complicated by the fact that some data are missing due to occlusions caused by different temporarily static objects (like cars, pedestrians etc.), self-occlusion of the façade due to an oblique viewing angle and variability in data resolution. So, in this paper we present a method that automatically detects and estimates windows on building façades while successfully addressing all these challenges.

Several previous works have attempted to tackle this problem of detecting window structures using 2D images (Mayer and Reznik, 2005, Ali et al., 2007, Lee and Nevatia, 2004). These methods either use explicit window models (Lee and Nevatia, 2004) along with horizontal and vertical analysis of images to discover repetitive patterns, or use an implicit window model and a learning approach to recognize windows (Mayer and Reznik, 2005, Ali et al., 2007). Compared to detection in 2D images, lesser work has been done on detecting windows from LiDAR data. In (Pauly et al., 2008), regularities of substructures are derived from a 3D model or range scan of a scene. This general approach may be used for extracting regularities but it is sensitive in the calculation of curvatures. In (Ali et al., 2008), a method is presented that converts LiDAR data into distance images, and then employs image processing techniques like morphological operations and contour analysis to segment windows. This 3D-to-2D conversion causes information loss. A Markov Network approach that requires training is used to label points as windows in (Triebel et al., 2006). In (Zheng et al., 2010), 3D repetitive elements are manually selected and automatically consolidated. In (Shen et al., 2011), façades are adaptively partitioned in horizontal and vertical planes based on the boundary features of the planar regions. However, the method works for data with uniform resolution, which is not always possible. In (Mesolongitis and Stamos, 2012), the 3D points are first projected onto a 2D binary orthographic point occupancy map, followed by the extraction and iterative refinement of the 2D window centers using a local lattice fitting and lattice voting scheme. These 2D centers are then projected back to 3D. This method, like most of the others, assumes regular symmetrical window patterns on the façade. However, these methods are less effective in case of lack of symmetry or large variations in symmetrical patterns (very common in case of interface of different façades joined together). Our method does not make any assumptions about the symmetry of the façade but in fact it also combines information from multiple successive passages to not only accommodate for lack of symmetry but also help to complete missing features due to occlusions. The property of windows not returning any signal is used and hence these windows are represented by holes on the 3D façade. A hole based extraction method is also presented in (Pu and Vosselman, 2007). The method searches edges along Triangular Irregular Network (TIN) of the façade to identify holes which are then fitted to rectangles. However, this bottom-up triangular meshing based method suffers from noisy LiDAR data and as it does not detect

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-3-1-2014

regular groups it is more prone to occlusions.

The proposed method first segments 3D points belonging to the building façade from the 3D urban point cloud. These 3D points are then projected onto a 2D plane parallel to the building façade. After point inversion within the watertight boundary, windows are segmented. The parameters/features of these segmented windows are then estimated exploiting symmetrically corresponding windows in the façade as well as temporally corresponding windows in successive passages, based on ANOVA (ANalysis Of VAriance) measurements. According to the best of our knowledge no prior work has ever been presented that fuses information from both symmetrical and temporal correspondences to detect and estimate windows. The estimated windows are then used to refine the 3D point cloud of the building façade via a reference table. An overview of the proposed method is presented in Algorithm 1.

2 SEGMENTATION OF BUILDING FAÇADES AND ROAD SURFACE FROM 3D URBAN POINT CLOUDS

In order to segment the building façades for window detection, we first classify the 3D urban point clouds into basic object types. Although several methods have been proposed for the classification of urban environments, we have used one of the more recent methods (Aijazi et al., 2013c) for this task. This method presents a super-voxel based approach in which the 3D urban point cloud is first segmented into voxels and then clustered into objects. Using local descriptors and geometrical features, these objects are then classified into 6 different classes including roads and buildings: {Road, Building, Car, Pole, Tree, Un-classified}. Some results of this method are shown in Fig. 1.

The occluded regions in each passage due to temporarily static and dynamic objects in the scene are completed with the help of similar 3D point clouds obtained from successive passages in the same place on different days and at different times of the day as presented in (Aijazi et al., 2013b) and described in Algorithm 1. In this method, the 3D point clouds in each successive passage are first registered with the former and the missing regions are completed using a point-to-point matching technique.

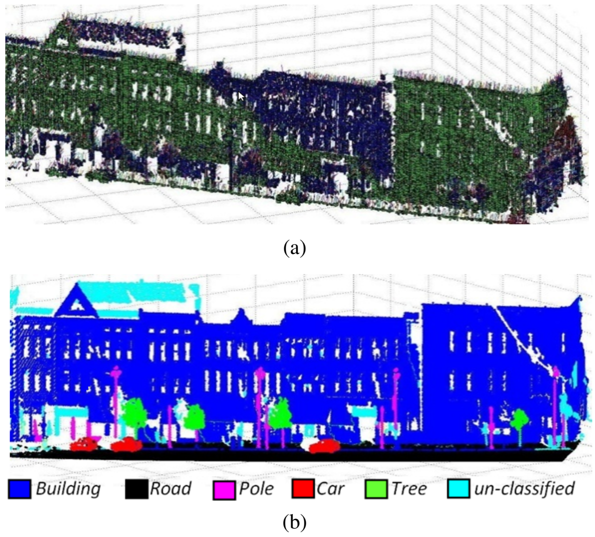
After classification, the 3D points pertaining to the segmented buildings as well as the road surface are then projected onto a plane, parallel to the building façade in order to detect and segment windows in the building façades as explained in the following section.

3 WINDOW SEGMENTATION

In the 3D LiDAR point clouds of the building façades, windows are usually represented by holes due to limited reflectivity (Mesolongitis and Stamos, 2012). They may, sometimes, also contain few 3D points with low intensity values due to reflections from objects inside (Tuttas and Stilla, 2011). Our method is based on this property. The 3D points pertaining to the building façade and road surface are first projected onto an arbitrary plane parallel to the building façade. A watertight boundary of this projected 2D façade is then determined. Point inversion is done within this boundary and windows are then segmented. The details of this method are as follows.

Algorithm 1 Automatic window detection and façade refinement.

```
Input: 3D urban point clouds $P(n_p)$ for passage number $n_p$
Segmentation based classification of $P(n_p)$ to obtain building façades and road surfaces
Register, match and compare point cloud $P(n_p)$ with $P(n_p - 1)$ to try to complete missing features on the building façade due to occlusions
Project the 3D points pertaining to building façades and road surfaces in $P(n_p)$ onto an arbitrary plane parallel to the building façade to obtain $Proj(B(n_p))$ and reference table $T_{proj}$
Extract a watertight boundary of the projected building façade in $Proj(B(n_p))$
Conduct point inversion within the watertight boundary to segment holes and obtain an inverted projection $Proj(B(n_p))_{INV}$
Cluster 2D points pertaining to holes in $Proj(B(n_p))_{INV}$ using a Nearest Neighbor Search (NNS)
Classify these segmented clusters as windows based on geometrical features and assign feature space {Height, Width, Profile, Geometrical center}
for all detected windows in the building façade do
    Find symmetrically corresponding windows in rows and columns on the façade
    Find temporally corresponding windows for the same façade from the previous passages in $Wc(n_p - 1)$
    Estimate window features using these correspondences
end for
Update the window features along with temporal correspondences in $Wc(n_p)$
if $n_p = n_{reset}$ then
    Reset $Wc$
end if
Correct $Proj(B(n_p))$ for estimated windows in $Wc(n_p)$
Refine and update 3D point cloud $P(n_p)$ using corrected $Proj(B(n_p))$ and $T_{proj}$
Delete $T_{proj}$
Delete $Proj(B(n_p))$ and $Proj(B(n_p))_{INV}$
Store $Wc(n_p)$
$Wc(n_p - 1) \leftarrow Wc(n_p)$
Store $P(n_p)$
$P(n_p - 1) \leftarrow P(np)$
return $P(n_p)$
```



(a)

(b)

Figure 1: (a) Voxel based segmentation of a particular scene. (b) Classified 3D points.

3.1 3D Point Projection

As a mobile street mapper, mentioned in Section 7, is used for data acquisition, the building class consists of 3D points predominantly belonging to building façades due to the particular viewing angle. These 3D points belonging to the building façade are then projected onto a known arbitrary plane parallel to the building façade in every passage n_p, as shown in Fig. 2(b). The projection plane is determined by calculating the best-fit plane through the 3D points of the building façade using planar regression of data as presented in (Fernández, 2005).

Although these 3D points are projected in 2D, a simple reference table T_{proj} is also generated that stores the number of the 3D points corresponding to the 2D points in the projection. This table T_{proj} only helps to correct and update the actual 3D point cloud once final processing is done (see Section 5 for details). It is eventually deleted.

3.2 Watertight Boundary Estimation

Now, in order to obtain a watertight boundary of the projected building façades, we first extract the outline/envelope of the façade using a sweep scan method presented in (Aijazi et al., 2013a). As the bottom part of the building outline close to the ground is often occluded and hence inconsistent due to the presence of different temporarily static or dynamic objects such as cars or pedestrians in the scene (Aijazi et al., 2013b), we use the 2D projection of the road surface to close the watertight boundary from below as shown in Fig. 2(c).

The 3D road points are projected onto the same arbitrary plane along with the building façade points as explained in Section 3.1. However, no reference table is generated. Now once the watertight boundary is extracted/obtained, we then perform a point inversion inside this boundary.

3.3 Point Inversion and Window Segmentation

As we are more interested in analyzing the holes in the building façade, we perform a point inversion within the watertight boundary. As a result of this point inversion, all holes including window holes surface out (a part of the segmentation process) as shown in Fig. 2(d) and we obtain an inverted projection.

The 3D points belonging to each of these holes are then grouped together by means of a standard agglomerative clustering algorithm using a distance based Nearest Neighbor Search (NNS) to obtain segmented clusters. This distance d is taken as $2 \times D_i$, where D_i is the size of the small regular interval used in the point inversion stage. These clusters are then classified as windows based on the geometrical sizes. Very small clusters (max width and height less than $0.5\ m \times 0.5\ m$ respectively) corresponding to small holes due to acquisition errors and imperfections are automatically removed. Similarly very large clusters (max width and height greater than $5\ m \times 5\ m$ respectively) corresponding to holes mainly due to occlusions, are also removed. The remaining clusters are then considered as windows as shown in Fig. 3(d). Each detected window is then defined by a feature space: $W_F \in \{H, W, P, GC\}$. Here H and W are the maximum height and width of the cluster respectively while GC is the geometrical center. P is a set of points that define the outline/profile of the window. These sets of points are calculated by using the same sweep scan algorithm (now in 2D) presented in (Aijazi et al., 2013a) on each window.

This step of point inversion not only reduces significantly the data size for analysis (few points corresponding to holes rather than those of the whole façade) but also allows easy segmentation of windows. Also, in case of any reflected points from within the window region, represented by small empty spaces between the window points after point inversion, they are also catered for as the whole region is merged to form a window cluster.

4 WINDOW FEATURES ESTIMATION

The windows detected in each passage are prone to different errors due to measurement errors, occlusions, etc. Hence, in order to estimate more accurately the detected windows along with their features/parameters, we then try to exploit both façade symmetry as well as multiple passages. In order to achieve this, we apply ANOVA method and try to estimate the window features using both symmetrical windows in the façade in each passage as well as the corresponding windows in the same façade in successive passages.

If no symmetrical correspondences (similar windows in symmetrical pattern) are found for a window in the façade, then it is only estimated by the corresponding window in successive passages. This way, we are able to detect and estimate windows and their features even for non symmetric building façades.

4.1 Exploiting Façade Symmetry

It is observed that usually building façades exhibit a structured arrangement consisting of repetitive patterns and self similarities. However, this is not always true due to rich architectural variations depending upon the type of buildings, countries and cultures.

In order to exploit the symmetrical properties of the façade (if possible), in each passage, we try to find symmetric correspondences (similar windows) on the building façade for each detected window. Symmetrical windows are found by simply analyzing the position of the windows on the façades by comparing the distance between their centers in both row and column. Windows at equidistant intervals in both row and column are considered symmetrical correspondences (at least 3 windows) and are grouped together as shown in Fig. 3. Symmetrical correspondences are determined for each window on the façade. In case of more than one type of symmetrical correspondences found for a particular window on the façade, the symmetrical correspondences corresponding to the smallest interval distances are considered.

Although symmetrically situated on the façade, their degree of similarity and how much these correspondences should contribute to estimate the features/parameters of a particular window are only ascertained after the ANOVA analysis.

4.2 Exploiting Multiple Passages

Once symmetrical correspondences for each window in the same façade are found in each passage, we then find corresponding windows in successive passages. As a result, we are able to estimate the window features more accurately, specially in case if we are not able to find any symmetrical correspondence due to either non symmetry or occlusions. Similarly, the redundant data obtained via several passages are also used to complete occluded regions of the façade (including windows) as explained in (Aijazi et al., 2013b). These passages (in the same place) could be on different days and at different hours of the day.

As the acquired 3D points are directly geo-referenced and then further co-registered in successive passages by applying ICP on

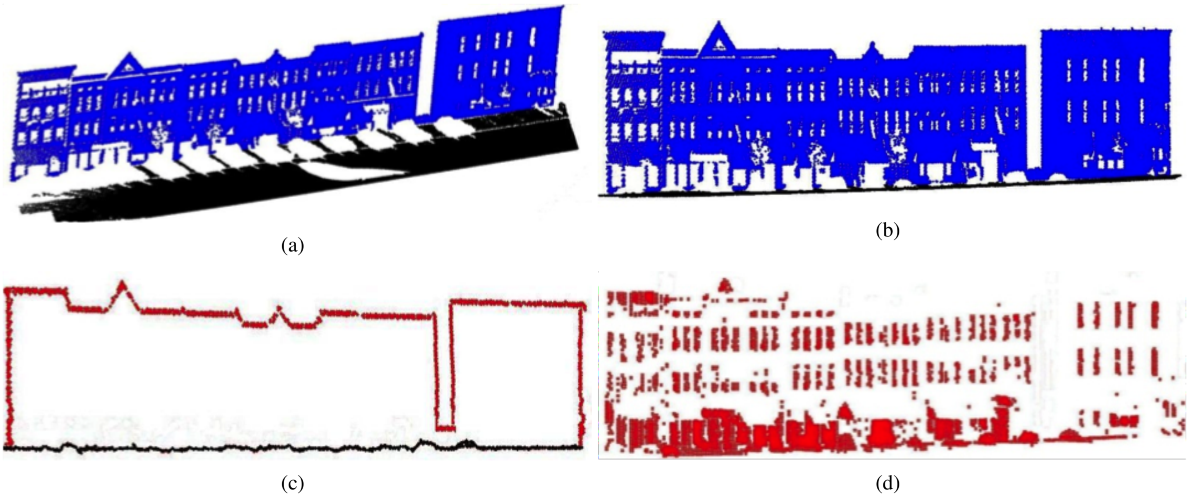

(a) (b) (c) (d)

Figure 2: (a) 3D points belonging to building (in blue) and road surfaces (in black). (b) 2D projection of the 3D points on an arbitrary plane. (c) Watertight boundary. In red building outline closed by road projection (black) from below. (d) After point inversion within the watertight boundary.

the skylines of the building as presented in (Aijazi et al., 2013b), there is no problem of finding corresponding windows for the same façade, in successive passages. Windows found in the same position on the façade in successive scans are considered as temporal correspondences and are grouped together. Temporal correspondences are determined for each window on the façade.

Although these temporal correspondences usually represent the same window, only after the ANOVA analyses we are able to ascertain how much these correspondences should contribute to estimate the features/parameters of a particular window.

4.3 ANOVA Based Estimation

In order to determine which correspondences should be used to estimate features/parameters for any particular detected window, we conduct a one way ANOVA (ANalysis Of VAriance) analysis (Tabachnick and Fidell, 2006) after adapting it to our requirements.

The statistic used in ANOVA partitions the variance into two components: 1) the between treatment variability, 2) the within treatment variability. Adapting them to our requirements, the within treatment variability SSW is used for symmetrical correspondences while between treatment variability SSB is used for temporal correspondences. It may be noted here that symmetrical correspondences are only considered for the current passage.

If n_w is the total number of detected windows (including both symmetric and temporal correspondences) in the total number of passages so far, G is the total number of passages so far, Y_{ip} is the value of Y^{th} feature for the i^{th} window in passage p, n_{wp} is the number of windows (only symmetrical correspondences) in passage p, $\overline{Y}_p$ is the mean of the feature values for all the windows (only symmetrical correspondences) in passage p, s_p^2 is the sample variance of the feature values for all the windows (only symmetrical correspondences) in passage p, and $\overline{Y}$ is the average of the feature values for all the windows (including both symmetric and temporal correspondences) in the total number of passages so far, then the total variabilities for the Y^{th} feature of the i^{th} window in passage p are given as:

$$SSW_{ip} = \sum_{p=1}^{G}\sum_{i=1}^{n_{wp}}(\overline{Y}_{ip} - Y_p)^2 = \sum_{p=1}^{G}(n_{wp} - 1)s_p^2 \quad (1)$$

Similarly,

$$\begin{aligned} SSB_{ip} &= \sum_{p=1}^{G}\sum_{i=1}^{n_{wp}}(\overline{Y}_p - \overline{Y})^2 \\ &= \sum_{p=1}^{G} n_{wp}(\overline{Y}_p - \overline{Y})^2 \\ &= \sum_{p=1}^{G} n_{wp}\overline{Y}_p^2 - \frac{(\sum_{p=1}^{G}\sum_{i=1}^{n_{wp}} Y_{ip})^2}{n_w} \end{aligned}$$

Further simplification yields:

$$SSB_{ip} = \sum_{p=1}^{G} n_{wp}\overline{Y}_p^2 - n_w\overline{Y}^2 \quad (2)$$

Using (1) and (2), we calculate the average amount of variability within each passage MSW and the average amount of variability between successive passages MSB as:

$$MSW_{ip} = \frac{SSW_{ip}}{df_{within}} \quad \text{and} \quad MSB_{ip} = \frac{SSB_{ip}}{df_{total}} \quad (3)$$

where df_{within} and df_{total} are the degrees of freedom given as $(n - G)$ and $(G - 1)$ respectively.

The F-statistic (Tabachnick and Fidell, 2006) is then calculated by computing the ratio of MSW and MSB to obtain:

$$F_{ip} = \frac{MSB_{ip}}{MSW_{ip}} \quad (4)$$

These values of F_{ip} and P_{ip} are calculated independently for both width and height features of each detected window in each passage. The maximum value of F_{ip} and P_{ip}, among the values obtained for both features, are then used for this analysis.

Now using df_{within} and df_{total} for each window, we determine the critical value F_c from the standard F-tables using $\alpha = 0.05$. Now using standard F-statistics, analyzing F_{ip} value of each window and comparing it with the corresponding F_c value helps us

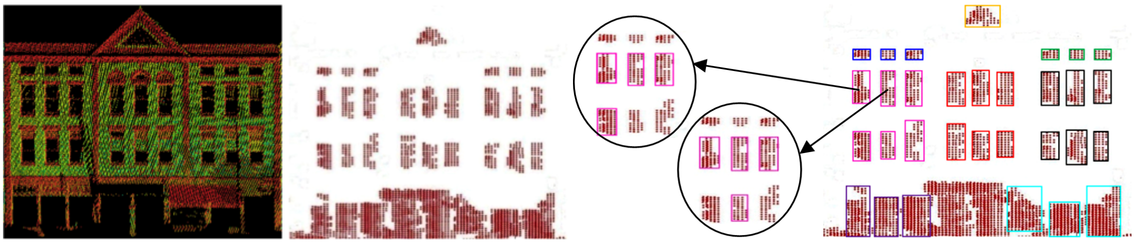


Figure 3: (a) 3D points of building façade. (b) Potential window candidates after point inversion and window segmentation. (c) Symmetrical correspondences in both row and column for window 1 and 2 respectively. Same colors (of the bounding box) represent similar windows.

to infer different weights to be used for the two types of correspondences (w_s and w_t for symmetrical and temporal correspondences respectively) for window feature estimation. There are 3 possible cases for the choice of these weights: 1) if $1 \leq F_{ip} \leq F_c$ then $w_s = w_t = 1$; 2) if $F_{ip} < 1$ then $w_t > w_s$; 3) if $F_{ip} > F_c$ then $w_s > w_t$.

As the null hypothesis being tested here is that windows in the symmetrical and temporal correspondences are similar, case 1 validates this hypothesis and hence both Symmetrical & Temporal correspondences are used to estimate the window features. Case 2 suggests that the symmetrical correspondences do not contain a lot of similar windows (due to occlusions or lack of symmetry in façade). Thus, in such a case, temporal correspondences are given more preference (higher weights) over symmetrical correspondences to estimate window features. Case 3 suggests that there are strong inconsistencies in the windows detected in successive passages. As a result, the variability of features of the detected window is much higher in successive passages as compared to the symmetrical correspondences found on the façade. This could be due to false detections, occlusions in certain passages or even changes occurring in the building façade (Aijazi et al., 2013a). Hence in this case, symmetrical correspondences are given more preference (higher weights) over temporal correspondences to estimate window feature.

Now in order to determine the weights associated with each correspondence for case 2 and case 3, we calculate the effect size measure ω^2 for ANOVA calculations. This measure, as expressed in (5) and (6), gives us the contribution of each of the two correspondences in the total variability observed.

$$\omega^2_{t_{ip}} = \frac{SSB_{ip} - (G-1)MSW_{ip}}{SSB_{ip} + SSW_{ip} + MSW_{ip}} \tag{5}$$

$$\omega^2_{s_{ip}} = 1 - \omega^2_{t_{ip}} \tag{6}$$

Here $\omega^2_{t_{ip}}$ and $\omega^2_{s_{ip}}$ are the contribution of the temporal and symmetrical correspondences respectively.

Once the weights for both correspondences are determined, the window feature value Y_{ip} (i.e. the Y^{th} feature for the i^{th} window in passage p) is estimated as $\hat{Y}_{ip}$:

$$\hat{Y}_{ip} = \frac{n_s\omega^2_{s_{ip}}(\sum_{j=1}^{n_t} Y_{t_j}) + n_t\omega^2_{t_{ip}}(\sum_{j=1}^{n_s} Y_{s_j})}{(n_s n_t)(\omega^2_{s_{ip}} + \omega^2_{t_{ip}})} \tag{7}$$

Here Y_t and Y_s are the values of the feature of the windows in temporal and symmetrical correspondences and n_t and n_s are the total number of windows considered as temporal and symmetrical correspondences respectively. Now using (7), both width and height features of each detected window in each passage are estimated independently.

Now, in order to estimate the profile P of the window in the feature space, we first find a set of common profile points for each of the two correspondences by sampling both height and width dimensions at small regular intervals as shown in Fig. 4. It should be noted that for symmetrical correspondences we first align the windows by shifting their centers.

Each point in the set of common profile points is then estimated by using (8) to obtain $\hat{P}_{ip} \in \{\hat{pt}_{ip_1} \dots \hat{pt}_{ip_k}\}$ such that:

$$\hat{pt}_{ip_k} = \frac{n_s\omega^2_{s_{ip}}(\sum_{j=1}^{n_t} pt_{t_{k_j}}) + n_t\omega^2_{t_{ip}}(\sum_{j=1}^{n_s} pt_{s_{k_j}})}{(n_s n_t)(\omega^2_{s_{ip}} + \omega^2_{t_{ip}})} \tag{8}$$

where pt_t and pt_s are the sets of common profile points corresponding to temporal and symmetrical correspondences while k is the k^{th} number of point in $\hat{P}_{ip}$, i.e. the estimated profile of the i^{th} window in passage p. Now using (7) and (8), the features of the window are estimated and then updated for that window along with the list of temporal correspondences in Wc as described in Algorithm 1.

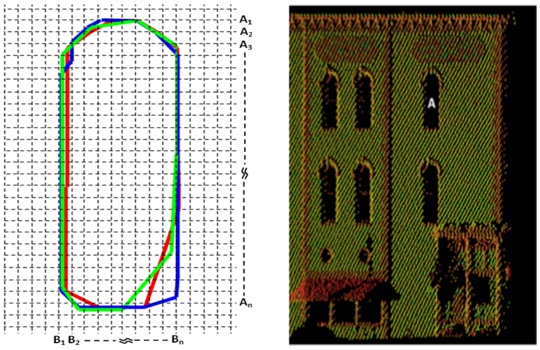
It must be noted here that in order to avoid initialization issues, for the first passage only symmetrical correspondences are considered whereas for all successive passages both symmetrical as well as temporal correspondences were considered in the equations for estimation.

5 REFINING/UPDATING THE BUILDING FAÇADES IN THE 3D POINT CLOUDS

Once the feature space of the detected windows is estimated and updated in Wc, the next step is to correct, refine and update the 3D façade in the actual 3D point cloud. In order to do this, we first correct the 2D projection by using the estimated profile of all the windows. These estimated profiles define the window space and hence all points within these profiles are considered as window points.

In many cases, the 3D point clouds obtained using LiDARs mounted on mobile terrestrial vehicles offer limited viewing angles and relatively sparse 3D points for distant windows. As a result, fine details pertaining to window panes, etc., within the window regions remain missing (see Fig. 4(b)). Hence, we prefer to delete the within window points to obtain well defined window holes in the 3D façades. On the other hand, if co-registered camera images are also available, then a 2D image of the window can be added to the window region (defined by the estimated profile boundary) to provide a more realistic/detailed view (Becker

doi:10.5194/isprsannals-II-3-1-2014



(a) (b)

Figure 4: (a) shows the profile points (connected with straight lines) of temporal correspondences for window "A" marked on façade in (b). In (a) red, blue and green color represent the profiles obtained in 1st, 2nd and 3rd passage respectively. A_n and B_n are the set of common profile points (along the two axis) obtained for the 3 profiles at nth regular interval (grid cell space). These points are then stored in pt_t, as these correspond to profiles of temporal correspondences.

and Haala, 2007). However, for most applications pertaining to accurate 3D urban modeling, geographical navigators, etc., well defined window-hole features on the building façade are more than sufficient.

Once the 2D points within the window region defined by the estimated window profile are deleted in the 2D projection, corresponding 3D points in the actual point cloud are also deleted using the reference table T_{proj} as shown in the results in Section 7. This processing ensures that the resulting building façade in the 3D urban point clouds contains accurate, well defined window regions.

6 AUTOMATIC CHECKS AND BALANCES

In case of any false detection, normally no symmetrical correspondences should be found on the building façade. However, in order to confirm whether this really is a false detection and not a non symmetric feature of the façade, we also check the temporal correspondences. In case no temporal correspondences are found, this detection is deleted from the list of windows Wc and is no longer used for refining the building façade. On the other hand if a certain detected window was found missing in any of the passages due to temporary occlusion, etc., it is still considered, as correspondences from other passages are used to estimate the window parameters and hence it remains in Wc and is also used for refining the building façades.

Similarly, we see that although effects of occlusions (both complete or partial) are greatly reduced due to the exploitation of multiple passages as explained in Section 2, sometimes certain window features remain partially or fully obscured due to window shutters or curtains, etc., in some passages, as shown in Fig. 6(a) (Zone marked in red). This may result in an inaccurate estimation of window parameters. However in our method, the increase in the variability of any of the correspondences, in such cases, results in its lower weighting in the parameter estimation of that window according to (7) and (8). This makes our method more robust and ensures a more accurate estimation of window parameters.

These checks and balances ensure that with the number of passages, window detection accuracy improves and the final building façade only contains the exact and accurate window features.

7 RESULTS, EVALUATION AND DISCUSSION

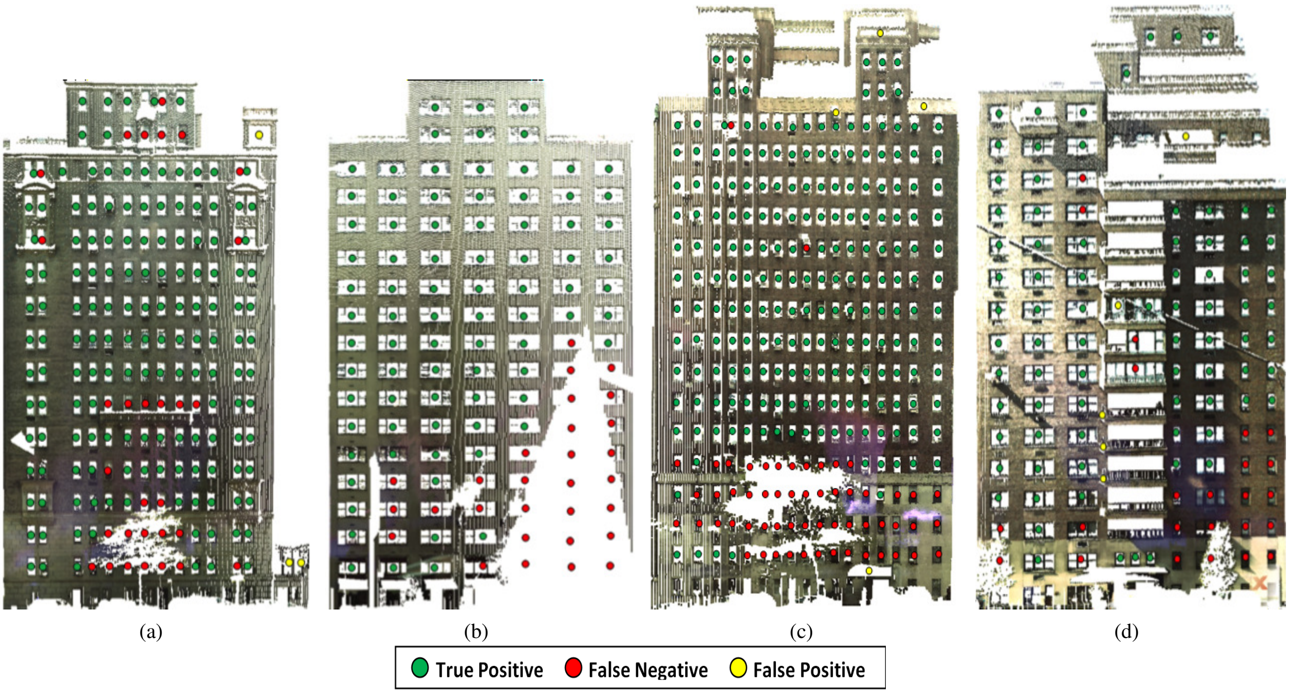
We first evaluated the window detection part of the proposed method using New York City data set presented and used by the authors in (Mesolongitis and Stamos, 2012). The data set contains a rich collection of 3D façades from the New York City area obtained from a single passage. Several building façades were used to evaluate our method. Some of the results are shown in Fig. 5. The results obtained (overall precision pr and recall re of 0.92 and 0.73 respectively) are comparable to the state of the art method (Mesolongitis and Stamos, 2012) evaluated on the façades of the same data set (pr 0.894, re 0.810). Our method received lower re scores compared to the state of the art method on façades with occlusions (as shown in Fig. 5 (a-c)) as the state of the art method predicted windows in the occluded regions based on symmetry of the façade compared to our method that exploits multiple passages (which were not available in this data set) to manage occlusions. These scores in our method improve considerably with multiple passages and exploiting temporal correspondences as discussed later on for the second data set. Also, contrary to our method, the state of the art method generated some false positives on certain façades with varying symmetry or in cases of interfacing façades. On the other hand, the relatively fewer false positives generated by our method are easily corrected exploiting temporal correspondences.

In order to fully evaluate our method, the dynamic data set of the 3D Urban Data Challenge 2011 was used. This data set contains 4 sets of the same dynamic scenes in downtown Lexington, Kentucky, USA obtained on different days and at different times (3D Urban Data Challenge 2011, 2011).

Figure 6 shows some qualitative results of window detection and refinement of different building façades in this data set. The method successfully detects and estimates windows of different types and sizes. Figure 6(a-d) illustrate how the method is able to successfully estimate window parameters in case of heavy occlusions (specially shop windows at ground level), where most other state of the art methods would probably fail as shown in (Lee and Nevatia, 2004). Figure 6(f) demonstrates that the proposed method successfully detects and estimates windows parameters pertaining to building façades with different non symmetric window patterns. As the method is non parametric (no assumption on window shapes) and does not rely solely on façade symmetry, it is able to estimate window parameters more accurately than conventional methods relying on regular patterns/façade symmetry only (Previtali et al., 2013).

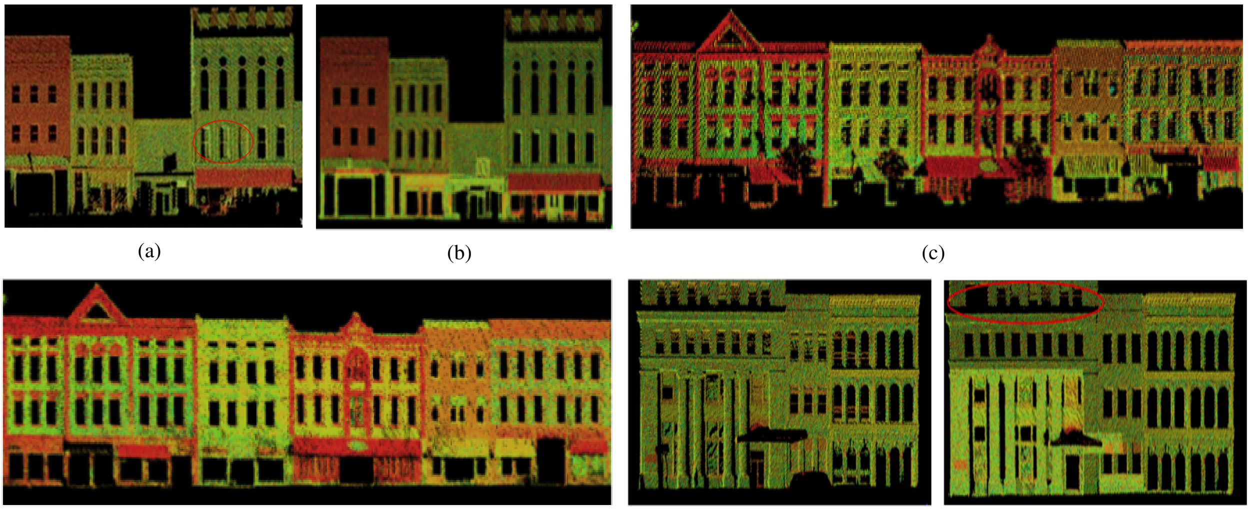
For quantitative analysis, we evaluated the accuracy of the window detection. A reset function was used to to avoid unnecessary memory storage issues, once the desired results are obtained. The n_{reset} value was fixed at 4 as the maximum available number of passages in the dataset was 4. However, where this reset can be done by human intervention, or manually set (as in our case) it could also be done automatically as presented in (Aijazi et al., 2013a). The window detection method was evaluated using different standard evaluation metrics as described in (Vihinen, 2012) (see Table 1). Ground truth was obtained by visual inspection of both raw 3D point clouds as well as the available 2D camera images of the building façades. The results clearly demonstrate the efficacy of the proposed method.

It is also observed that the **PPV** is generally found higher than **NPV**, which suggests that the method is more conservative and is more likely to detect and estimate windows once highly certain,



(a) (b) (c) (d)

Figure 5: Window detection results are shown for some façades. In (a), (b) & (c) we can see a number of false negatives due to occlusions. In (c) & (d) we also find some false negatives due to closed windows because of window shutters or curtains.



(a) (b) (c)

(d) (e) (f)

Figure 6: (a) & (c) show raw 3D points (color coded intensity) of different building façades from the data set containing windows of different types and sizes while (b) & (d) are the corresponding 3D point clouds after window detection and refinement. (b) and (d) clearly demonstrate the efficacy of the method in case of heavy occlusions specially at ground level shop windows. (e) shows the raw 3D points of a building façade with different non symmetric window patterns. In (f) the same façade after successful window detection and refinement can be seen.

Table 1: Window Detection Evaluation.

ACC	Accuracy	0.917
PPV	Positive Predictive Value	0.971
NPV	Negative Predictive Value	0.908
FDR	False Discovery Rate	0.029
F_1	F_1 measure	0.912

otherwise it does not. Hence, false positives usually occur less often than false negatives (also evident in Fig. 5).

Although the method successfully detects windows of different shapes and sizes, and estimates their parameters even in difficult circumstances such as temporary occlusions and non symmetrical façade structures, it has some limitations in case of permanent occlusions resulting in permanently missing 3D points, for example part of the building façade (zone marked in red) in Fig. 6(f). This permanently occluded region, due to the limited viewing angle of the data acquisition vehicle, results in large, deformed and interconnected window segments after point inversion and hence are not considered as a window nor estimated. Sometimes smaller permanent occlusions could also result in false positives. In such conditions, contrary to our algorithm, a classical symmetry based window detection algorithm can assume/predict (which may be wrong) windows in the region based on the overall symmetry of the façade.

8 CONCLUSION

In this paper a method has been presented that automatically detects windows and estimates their parameters in 3D LiDAR point clouds obtained from mobile terrestrial data acquisition systems in the urban environment. The main feature of this method is that it fuses information from both symmetrical and temporal correspondences using ANOVA measurements. This unique fusion of information not only accommodates for lack of symmetry but also helps complete missing features due to occlusions. These estimated windows are then used to refine 3D building façades for different applications pertaining to accurate 3D urban modeling or geographical navigators that frequently update their database.

The results evaluated on real data sets using different standard evaluation metrics clearly demonstrate the technical prowess of the method.

ACKNOWLEDGMENTS

This work is supported by the Agence Nationale de la Recherche (ANR - the French national research agency) (ANR CONTINT iSpace&Time – ANR-10-CONT-23) and by “le Conseil Général de l’Allier”.

REFERENCES

3D Urban Data Challenge 2011, 2011. www.vis.uky.edu/3dimpvt/challenge.php.

Aijazi, A. K., Checchin, P. and Trassoudaine, L., 2013a. Automatic Removal of Imperfections and Change Detection for Accurate 3D Urban Cartography by Classification and Incremental Updating. Remote Sensing 5(8), pp. 3701–3728.

Aijazi, A. K., Checchin, P. and Trassoudaine, L., 2013b. Automatic Update and Completion of Occluded Regions for Accurate 3D Urban Cartography by Combining Multiple Views and Multiple Passages. In: Int. Conf. on Pattern Recognition, Applications and Methods, Barcelona, Spain.

Aijazi, A. K., Checchin, P. and Trassoudaine, L., 2013c. Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach. Remote Sensing 5(4), pp. 1624–1650.

Ali, H., Ahmed, B. and Paar, G., 2008. Robust Window Detection from 3D Laser Scanner Data. In: CISP 2008 Congress on Image and Signal Processing, May 28-30 2008 Sayna/Hainan, China, Vol. 2, pp. 115–118.

Ali, H., Seifert, C., Jindal, N., Paletta, L. and Paar, G., 2007. Window Detection in Façades. In: Image Analysis and Processing. 14th International Conference on, pp. 837–842.

Becker, S. and Haala, N., 2007. Refinement of Building Façades by Integrated Processing of LiDAR and Image Data. In: Proceedings of ISPRS Technical Commission III Symposium ”Photogrammetric Image Analysis”, Vol. 36 (Part 3/W49A), Munich, Germany, pp. 7–12.

Fernández, Ó., 2005. Obtaining a best fitting plane through 3d georeferenced data. Journal of Structural Geology 27(5), pp. 855 – 858.

Lee, S. C. and Nevatia, R., 2004. Extraction and Integration of Window in a 3D Building Model from Ground View Images. In: Computer Vision and Pattern Recognition. Proc. of the 2004 IEEE Computer Society Conference on, Vol. 2, pp. II–113.

Mayer, H. and Reznik, S., 2005. Building façade interpretation from image sequences. In: ISPRS Workshop CMRT, Vol. 2005.

Mesolongitis, A. and Stamos, I., 2012. Detection of windows in point clouds of urban scenes. In: CVPR Workshops, IEEE, pp. 17–24.

Pauly, M., Mitra, N. J., Wallner, J., Pottmann, H. and Guibas, L., 2008. Discovering Structural Regularity in 3D Geometry. ACM Transactions on Graphics 27(3), pp. #43, 1–11.

Previtali, M., Scaioni, M., Barazzetti, L., Brumana, R. and Roncoroni, F., 2013. Automated detection of repeated structures in building facades. In: Laser Scanning 2013 ISPRS Workshop, Vol. II (Part 5/W2), ISPRS Annals PRS&SIS, Antalya, Turkey, pp. 241–246.

Pu, S. and Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. In: IAPRS & SIS, Vol. XXXVI (Part 3/W52), Espoo, Finland.

Shen, C.-H., Huang, S.-S., Fu, H. and Hu, S.-M., 2011. Adaptive Partitioning of Urban Façades. ACM Trans. Graph. 30(6), pp. 184:1–184:10.

Tabachnick, B. G. and Fidell, L. S., 2006. Using Multivariate Statistics (5th Edition). Allyn & Bacon, Inc., Needham Heights, MA, USA.

Triebel, R., Kersting, K. and Burgard, W., 2006. Robust 3D scan point classification using associative Markov networks. In: Robotics and Automation (ICRA). Proc. IEEE Intl Conf. on, pp. 2603–2608.

Tuttas, S. and Stilla, U., 2011. Window detection in sparse point clouds using indoor points. In: IAPRS & SIS, Vol. XXXVIII (Part 3/W22), Munich, Germany.

Vihinen, M., 2012. How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. BMC Genomics 13(Suppl 4), pp. S2.

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N. J., Cohen-Or, D. and Chen, B., 2010. Non-local Scan Consolidation for 3D Urban Scenes. ACM Trans. Graph. 29(4), pp. 94:1–94:9.