Robust Reconstruction of Building Façades for Large Areas Using Spaceborne TomoSAR Point Clouds

Muhammad Shahzad(1), Student Member, IEEE, and Xiao Xiang Zhu(1,2), Member, IEEE

(1) Technische Universität München, Lehrstuhl für Methodik der Fernerkundung, Arcisstrasse 21, 80333 Munich, Germany
(Email: muhammad.shahzad@bv.tum.de, Tel: +49-81-5328-3096)

(2) German Aerospace Center (DLR), Remote Sensing Technology Institute (IMF), Oberpfaffenhofen, 82234 Wessling, Germany (Email: xiao.zhu@dlr.de, Tel: +49-81-5328-3531)

Abstract

With data provided by modern meter-resolution SAR sensors and advanced multi-pass interferometric techniques such as tomographic SAR inversion (TomoSAR), it is now possible to reconstruct the shape and monitor the undergoing motion of urban infrastructures on the scale of centimeters or even millimeters from space in very high level of details. The retrieval of rich information allows us to take a step further towards generation of 4-D (or even higher dimensional) dynamic city models, i.e. city models that can incorporate temporal (motion) behaviour along with the 3-D information. Motivated by these opportunities, the authors proposed an approach that first attempts to reconstruct façades from this class of data. The approach works well for small areas containing only a couple of buildings. Yet towards automatic reconstruction for the whole city area, a more robust and fully automatic approach is needed. In this paper, we present a complete extended approach for automatic (parametric) reconstruction of building façades from 4-D TomoSAR point cloud data and put particular focus on robust reconstruction of large areas. The proposed approach is illustrated and validated by examples using TomoSAR point clouds generated from a stack of TerraSAR-X high-resolution spotlight images from ascending orbit covering approx. 2 km² high rise area in the city of Las Vegas.

Keywords: façade reconstruction, 4-D point cloud, TerraSAR-X, tomographic SAR inversion, clustering, point density.
Automatic detection and reconstruction of buildings has been an active research area for at least two decades. Despite an extensive research effort, the topic is still of great interest due to ever increasing growth of urban population which gives rise to wide range of potential applications in various fields. For instance, building footprints (i.e., 2-D building outline/shape) can be used for urban landscapes development [1], urban planning [2], damage assessment [3], disaster management [4], navigation purposes [5] etc. Additionally, 2-D footprint data in combination with height information can generate 3-D building models. Such models are essential for virtual city modeling [6] and 3-D GIS applications (e.g., commercial softwares such as Google Earth, Apple Maps etc.). Other possible usages may include analyzing solar potential over building roofs [7], placing and installing telecommunication antenna towers[8], web based mapping [9], tourism [6], architecture [10], augmented reality applications [5][11] and many more.

Spaceborne synthetic aperture radar (SAR) sensors are able to provide day/night global coverage in virtually all weather conditions. Moreover, due to coherent imaging nature, by using acquisitions taken at different times, it is also uniquely capable of imaging the dynamics of the illuminated area in the scale of centimeters or even millimeters. These benefits have motivated many researchers and therefore several methods have been developed that use very high resolution (VHR) spaceborne SAR imagery for detection and reconstruction of man-made structures in particular buildings. For instance, single-aspect SAR images based approaches for building reconstruction are proposed in [12]-[14]. Due to the fact that only single SAR images are used, these approaches predominantly perform well mostly only for isolated buildings, but not for dense urban areas where the buildings are densely packed and smaller buildings are often occluded by the higher ones[15]. To resolve this, interferometric SAR (InSAR) data, SAR image pairs taken from slightly different viewing angles, are used. E.g., a modified machine vision approach is proposed in [16] to detect and extract buildings. The algorithm is based on local approximation of best fitting planes in the segmented regions of interest. Similarly, Thiele et al.[17] also developed a model based approach to detect and reconstruct building footprints using orthogonal InSAR images. Another automatic approach based on modeling building objects as cuboids using multi-aspect
polarimetric SAR images is presented in [18]. In data fusion context, the use of optical imagery has also been exploited along with SAR [19] and InSAR [15] datasets, respectively. Despite of the active ongoing research in the area, the problem of building reconstruction still remains challenging due to inherent characteristics with SAR images such as geometrical projection caused by the side-looking geometry [20]. Moreover, complex building structures and high variability of objects appearing in the images make automatic building detection and reconstruction a challenging task. For example, problems posed by occlusion of smaller buildings/objects from the higher ones render difficulties in large area extension. Therefore, prior knowledge is often incorporated with certain regularization (geometric) constraints for realistic and automatic reconstruction. For instance, façades are often assumed to be vertical [19], building footprint as regular parallelepipeds [14] and roofs as polyhedral structures [21] etc.

Modern spaceborne SAR sensors such as TerraSAR-X/TanDEM-X [22] and COSMO-SkyMed [23] can deliver VHR resolution data that fits well to the inherent spatial scales of buildings. Hence, this VHR data is particularly suited for detailed urban mapping [24]-[33]. However, 2-D SAR imaging projects the 3-D scene onto a 2-D image making it "non-injective" in urban scenarios due to the presence of vertical structures (e.g., building façades or other man-made objects) [25]. The unwelcoming effects such as layover and foreshortening seriously handicap the interpretation of SAR images. Advanced interferometric techniques, such as persistent scatterer interferometry (PSI) and SAR tomography (TomoSAR), aim at SAR imaging in 3-D or even higher dimensions. Among them, PSI exploits the coherent pixels i.e., the bright long term stable objects (persistent scatterers) [30]. However it assumes single scatterers in one azimuth-range pixel and therefore does not resolve the layover problem. TomoSAR, on the other hand, aims at real and unambiguous 3-D SAR imaging [34]-[36][25]. By exploiting multi-pass SAR images taken from slightly different positions, like PSI does, it builds up a synthetic aperture in the elevation direction that enables retrieval of precise 3-D position of dominant scatterers via spectral analysis within one azimuth-range SAR image pixel [25]. Multiple layovered objects in any pixel are therefore separated from the reconstructed reflectivity profile in elevation direction [36]. Moreover, exploiting the fact that different acquisitions are taken at different times, the
synthetic aperture can also be extended in the temporal domain to enable 4-D (space-time) focusing of SAR images. The technique referred to as D-TomoSAR which combines the strengths of both TomoSAR and PSI [37][38][27][26][39][40]. It is capable of retrieving elevation and deformation information (linear, seasonal, etc.) even of multiple scatterers inside a single SAR image pixel [25][29]. Retrieval of rich scatterer information from VHR D-TomoSAR enables generation of 4-D (space-time) point cloud of the illuminated area with point (scatterer) density that is comparable to LiDAR. E.g. experiments using TerraSAR-X high-resolution spotlight data stacks show that the scatterer density retrieved using TomoSAR is in the order of 1 million pts/km² [28].

Object reconstruction from these high quality TomoSAR point clouds can greatly support the reconstruction of dynamic city models that could potentially be used to monitor and visualize the dynamics of urban infrastructure in very high level of details. Such models would be immensely helpful to ensure safety/security of growing urban population by monitoring of urban infrastructures against potential threats of damage and structural degradation caused by various factors e.g., ground subsidence or uplift, bad construction, natural disaster etc. Motivated by this, the very first results of building façade reconstruction from single view (ascending stack) and multi-view (fused ascending and descending stacks) perspectives over a small test building area (Bellagio hotel, Las Vegas) were presented in [41] and [42] respectively.

In this paper, we present an approach that allows automatic reconstruction of 3-D building façades using these unstructured TomoSAR point clouds only. The approach proposes new methods as well as modifications to the previously introduced idea in [42] aimed at finding a more general solution towards automatic reconstruction of the whole city area. The basic idea is to reconstruct 3-D building models via independent modeling of each individual façade to build the overall 2-D shape of the building footprint followed by its representation in 3-D. Following are the innovative contributions that are specific to the approach proposed in this paper:

- A robust M-estimator based directional SD estimation method is proposed which provides much better estimates of façade regions compared to the grid based SD estimation proposed in [42] by incorporating the façade geometry. Moreover, instead of rejecting non-façade points via 2-D
morphological operations used in [42], robust 3-D surface normals information is utilized. Use of additional dimensional with the vertical façade assumption helps in better rejecting non-façade points;

- K-means clustering with a criterion for prior guessing the number of clusters is used in previous works [41][42]. This technique provides good results for single buildings but when it comes to larger areas, there are two major concerns: 1) guessing number of clusters is not always trivial; 2) certain shape of clusters are not very well recognized. For this reason, a three-step automatic (unsupervised) clustering approach that combines both the density based clustering [43] and mean shift algorithm [44][45] is proposed in this paper. The proposed segmentation approach is able to work directly on bigger areas without requiring any prior knowledge about the number of clusters.

- Façades are modeled using general (1st and 2nd order) polynomial equations to cater for wide variety of buildings footprint. Detailed methodological description of the modeling procedure is explained that is able to cater arbitrary (rotated) orientations of building façades. The coefficients of the model are estimated using weighted total least squares (WTLS) method to cope for localization errors of TomoSAR points in both $xy$ directions.

- During the reconstruction procedure, the presence of smaller clustered segments occurring at façade transition regions handicaps accurate determination of vertex points from the adjacent façade pair and may cause the reconstruction procedure to fail. To deal with this problem, smaller "conflicting segments" are automatically identified and removed for accurate and robust reconstruction of the adjacent façades.

- Side-looking SAR geometry and complexity of the scene in dense high rise area of interest can cause occlusions of lower height façades scattered around higher building façade structures. As a consequence, few or sometimes no data is available for the occluded region rendering incomplete reconstruction or breaking an individual façade into two or more segments. A partial solution is also presented in this paper which refines the reconstructed façade footprints via insertion (of additional segments) and extension operations.
Lastly, the paper presents the first demonstration of automatic large area reconstruction of building façades from this class of data. Moreover the developed methods are not strictly applicable to TomoSAR point clouds only but are also applicable to work on unstructured 3-D point clouds generated from a different sensor with similar configuration (i.e., oblique geometry) with both low and high point densities.

The aforementioned contributions allow completely automatic (but parametric) reconstruction of building façades from TomoSAR point clouds in larger areas.

The remainder of the paper is structured as follows: Section II presents a brief procedural overview of the existing techniques that use 3-D LiDAR point cloud for building reconstruction. Section III presents in detail the proposed approach. In Section IV, the experimental results, obtained from the TomoSAR point cloud generated from a TerraSAR-X high resolution spotlight data stack (ascending orbit only), are presented and discussed. Finally, in Section V, some conclusions about the proposed approach are drawn and future perspectives are discussed.

II. RELATED WORK

Most approaches employ airborne LiDAR data for automatic 3-D reconstruction of buildings. Methodologically, the problem is tackled by subdividing the task into two sequential steps, i.e., detection/classification of building points followed by their 3-D modeling and reconstruction.

Detection is generally carried out by first computing the digital terrain model (DTM) by filtering techniques, e.g., morphological filtering [46], gradient analysis [47], or iterative densification of triangular irregular network structures [48]. Nadir looking LiDAR points essentially gives a digital surface model (DSM). Subtracting DSM from the computed DTM provides a normalized DSM (nDSM) which represents the height variation of non-ground points. Subsequently, building points are extracted out from nDSM by exploiting geometrical features such as deviations from surface model, local height measures, roughness and slope variations. Methods based on building boundary tracing from nDSM [49] or directly from point clouds [50][51] have also been employed for building detection. With them, finer building boundaries are determined by regularization of the coarsely traced boundaries. All points that
lie inside the boundary regions are considered as building points. Building points can also be extracted out by explicitly labeling every point in the data set. For labeling purpose, local neighborhood features such as height, eigenvalue and plane features have been used in conjunction with supervised [52], semi supervised [21] and unsupervised [53] classification techniques.

Detected building regions or points are in turn used for 3-D modeling and reconstruction. Most methods make use of the fact that man-made structures such as buildings usually have either parametric shapes (model driven) or composed of polyhedral structures only (data-driven). The latter is however more common in the literature where local sets of coplanar points are first determined using 3-D Hough transform or RANSAC algorithms and then reconstruction is carried out by surface fitting in the segmented building regions followed by region growing procedure [53] or by building up an adjacency graph [21][54].

The above mentioned methods and the majority of other techniques present in the literature that make use of 3-D LiDAR data cannot be directly applied to TomoSAR point clouds due to side looking SAR geometry and different microwave scattering properties of the objects appearing in the scene reflecting different geometrical and material features. Compared to LiDAR, TomoSAR point clouds possess following peculiarities that should be taken into consideration:

**Accuracy & errors**

- TomoSAR point clouds deliver moderate 3-D positioning accuracy in the order of 1m [15] as compared to (airborne) LiDAR systems having accuracy in the order of 0.1m [21];
- Ghost scatterers [55] may be generated due to multiple scattering that appear as outliers far away from a realistic 3-D position;
- Small number of images and limited orbit speed render location error of TomoSAR points highly anisotropic with an elevation error typically one or two orders of magnitude higher than in range and azimuth [25].

**Coherent imaging**

- Due to coherent imaging nature, temporally incoherent objects such as trees cannot be reconstructed from multi-pass spaceborne SAR image stacks.
Side looking geometry
- Separation of layover on vertical structures render geo-coded TomoSAR point clouds to possess higher density of points on building façades;
- In order to obtain full structure of individual buildings from space, multiple views are required [42].

Higher dimensional imaging
- In addition to 3-D spatial information, TomoSAR point clouds also possess the fourth dimensional information, i.e., temporal or seasonal deformation estimates, making them very attractive for dynamic city modeling.

III. PROCESSING STEPS FOR BUILDING FAÇADES RECONSTRUCTION

Due to the side-looking SAR geometry, when projected the TomoSAR point clouds onto ground plane vertical façade regions exhibit higher scatterer (point) density (SD) as compared to non-façade regions. It is mostly true due to the existence of strong corner reflectors, e.g., window frames on the building façades. Taking this fact into account, in [42], we proposed to extract façade points by projecting all the points onto xy grid for estimating SD (rastered image) followed by thresholding and applying morphological dilation operation. This approach works well for high rise buildings having much higher point density but limits the extraction of façade points from relatively lower buildings. The selection of a particular threshold thus becomes crucial.

To resolve this issue, in this work a more robust façade extraction approach is proposed which is based on directional SD estimation procedure to locally estimate the SD for each point while incorporating the façade geometry [56]. Later, robust 3-D surface normals information is utilized to extract façade points. Automatic segmentation of extracted façade points is obtained by first performing coarse clustering to cluster points belonging to individual buildings. Then each cluster is further fine clustered using Gaussian map based mean shift clustering algorithm. After that, clusters within clusters are spatially separated. Segmented façades are then classified as flat or curved and their model parameters are estimated. Subsequently, the geometric primitives such as vertex points are determined.
from the intersection of adjacent façade pair after removing smaller conflicting segments occurring at transitional regions. Finally, a refinement operation is carried out on the reconstructed façades that remain either incomplete or broken into two or more segments to complete the reconstruction process.

Figure 1: Block diagram of the proposed method

Figure 1 shows the block diagram of the processing steps involved in the complete methodology. Next we explain in detail the procedures of the proposed approach in dedicated subsections.
A. SD estimation

For each 3-D TomoSAR point $p$, points within its local neighborhood $v_c$ are used for SD estimation. $v_c$ includes all those points that lie inside a vertical cylinder centered at $p$. To emphasize the building façades, we incorporate façade geometry in estimating SD. I.e., we estimate direction of the local neighborhood via line fitting using robust M-estimator. The method iteratively reweights the points according to the residuals and computes the so-called M-estimates as follows [57]:

1. Initial estimates of the line parameters $\beta$ (e.g., $\beta_1 = \text{slope}$ and $\beta_2 = \text{offset}$) are derived from ordinary least squares;
2. Weights $w_{p_i}$ of each point $p_i \in v_c$ are then computed using a bisquare function [58][59]:
   
   $$w_{p_i} = \begin{cases} 
   \left(1-u^2\right)^2 & \text{for } \text{abs}(u) < 1 \\
   0 & \text{otherwise} 
   \end{cases}$$

   where $u = \frac{\left|y_{p_i} - x_{p_i}\beta_1 - \beta_2\right|}{4.685\hat{\sigma}\sqrt{1-t}}$

   where $t$ is the leverage computed using parameter estimates $\beta$ of the fitted line and $\hat{\sigma}$ is the estimate for the scale of the error term computed by $\hat{\sigma}=1.483\*\text{MAD}$ where MAD is the median absolute deviation of the residuals from their median. The term 1.483 is used to make the estimate $\hat{\sigma}$ consistent to standard deviation at Gaussian distribution [60][58].
3. Updating $\beta$ using weights $w_{p_i}$ by applying weighted least squares to solve the following objective function:
   
   $$\arg \min_{\beta} \sum_{p_i \in v_c} w_{p_i} \left(\beta\right) \left|y_{p_i} - x_{p_i}\beta_1 - \beta_2\right|^2$$

   where $x_{p_i}$ and $y_{p_i}$ represent the abscissa and ordinate (i.e., ground coordinates) of the points within $v_c$;
4. Iterating steps 2 and 3 until a fixed number of iterations.
The estimated line describes the main principal axis of the cylindrical footprint of the local neighborhood. Orthogonal distance for every point in $v_c$ is then calculated from the principal axis (shifted to the point $p$) and the points having distances less than $d$ are taken as “inliers” and used in $SD$ estimation.

$SD$ for each point is thus defined as the number of points within a directional (cylindrical) neighborhood window divided by the area of the window:

$$SD = \frac{\text{number of points in } v_d}{\text{Area of } v_d}$$

(3)

where $v_d \subseteq v_c$ but includes only those points that lie close to the principal axis of points in $v_c$.

Figure 2 shows the graphical representation of the $SD$ estimation procedure:

![Figure 2: Illustration of $SD$ estimation: (a) 3-D view of the local cylindrical neighborhood around the point of interest; (b) Top view of (a). The coefficients of the dotted yellow regression line are estimated via M estimation. Black dotted line shows the shift of yellow line to the point of interest. Shaded area shows the region of $v_d$ within $v_c$.](image)

**B. Façade extraction**

Based on the estimated $SD$, façade points can be extracted. For large area, both high and low buildings are present. A hard threshold, i.e. removing points below a rather high $SD$ value, as used in [42], would lead to miss-detection of façade points of lower buildings whose $SD$ estimates would be
relatively low with respect to high rise buildings. To avoid such miss-detection, we extract façade points in a sequential way. Firstly, we have lowered the $SD$ threshold to detect not only higher buildings but also lower ones. However, a softer threshold would also introduce false positives, i.e. roof points or ground points with a local point density comparable to those of lower buildings. Therefore, we introduce a second step which utilizes 3-D surface normals information by incorporating prior knowledge (i.e., façades are assumed to be vertical surfaces which separate them from non-vertical ground plane and roofs) to eliminate those false positives.

The key issue is then the local surface normal estimation for each 3-D point. Commonly, they are estimated via fitting “best” plane in least squares (LS) sense in the local neighborhood $v_c$ which is equivalent to performing principal component analysis (PCA) of the points in $v_c$ [61]. This implies that the surface normals can be directly estimated for each 3-D point via eigenvalue/eigenvector analysis of 3-D (i.e., 3x3) covariance matrix $\Sigma_{v_c}$. There are two advantages to use eigenvalue/eigenvector analysis of $\Sigma_{v_c}$ for surface normals estimation:

- First, the eigenvector associated to the smallest eigenvalue of the positive semi-definite matrix $\Sigma_{v_c}$ is the direct estimate of the local surface normal of the query point [61];

- Secondly, in addition to giving us the direct solution of estimating local surface normal, it can also help us in determining the dimensionality of each 3-D point [21]. To elaborate, eigenvectors of $\Sigma_{v_c}$ essentially give the orthonormal basis for the local neighborhood $v_c$ with their corresponding eigenvalues representing the magnitude (or variance) of expansion. Analyzing this magnitude implicitly give us an indication of the beneath surface. E.g., eigenvalue analysis of $\Sigma_{v_c}$ with all points lying on the plane would ideally return only two nonzero eigenvalues. Similarly for all points lying on a 3-D line would give only one nonzero eigenvalue. Eigenvalue analysis for segmentation and classification of planar points has been exploited in [21][52][53].

Eigenvalue/eigenvector analysis via classical PCA may fail to give precise estimate of the 3-D surface normal using TomoSAR point cloud due to presence of outliers and localization errors (see
section II: accuracy and errors). Robust estimation of the covariance matrix $\Sigma_v$ is therefore needed. To this end, we estimated $\Sigma_v$ using robust minimum covariance determinant (MCD) method [62]. The method finds a subset (fraction) $\alpha$ of the data points $p_i \in v_c$ whose covariance matrix has the lowest determinant. The idea stems from the concept of generalized variance (GV) which is defined to be the determinant of the covariance matrix of any $d$-D ($d>1$) random variable [63]. For instance, in case of 2-D ($x$-$y$) points, the GV provides a scalar value which measures the overall variability of all points in both $x$ and $y$ dimensions. Points that are clustered tightly together tend to have smaller GV (i.e., lower determinant of their covariance matrix) as compared to scattered ones. Thus the subset $\alpha$ of the data points which provides the lowest determinant is taken as MCD estimate of $\Sigma_v$. If the data points are assumed to have less than 25% outliers, then an appropriate selection of $\alpha = 0.75$ (also used in this work) proposed in [64] provides good compromise between statistical efficiency and high breakdown value ($\alpha = 0.75$ implies that 75% of the data points have been used in covariance estimation).

The covariance matrix $\Sigma_v$ estimated using MCD method from the local neighboring points $p_i \in v_c$ around (in cylinder) the point of interest $p_o(x_o,y_o,z_o)$ is then used to determine the local 3-D surface normal at $p_o$. If we denote a plane which robustly fits the neighboring points $p_i$ as $n_x x + n_y y + n_z z + \rho = 0$ with $\rho = -n_x x_o - n_y y_o - n_z z_o$, then $N_o(n_x, n_y, n_z)$ depicts the local 3-D surface normal at $p_o$. $N_o$ is thus directly estimated from $\Sigma_v$ by computing the eigenvector associated to the smallest eigenvalue of $\Sigma_v$ (here $v_c$ includes points in the vicinity of $p_o$) i.e.,

$$\begin{align*}
\text{if } \Sigma_v \cdot v_j &= \lambda_j \cdot v_j, \quad j = 1, 2, 3 \text{ (descending order)} \\
\text{then surface normal of the underlying surface at point } p_o: N_o(n_x, n_y, n_z) &= v_3
\end{align*}$$

(4)

From (4), robust 3-D surface normals are computed for each point that is obtained after SD thresholding. Ideally, the direction of surface normal should be parallel to the ground for points on the vertical façades which separate them from non-vertical ground plane and roofs. Taking this fact into account, façade points are extracted out by retaining only those points having normals are close to the
horizontal axis (i.e., parallel to ground for points belonging to a vertical surface). In this manner, the proposed two step approach allows us to robustly extract façade points over a large area where both high and low buildings are present.

Figure 3 shows a comparison of the proposed approach with the one presented in [42]. The selected area shown in Figure 3(a) contains relatively lower height buildings with low and inconsistent density of points on building façades. It can be seen that in comparison to the SD estimation results from the previous approach depicted in Figure 3(b), higher and complete density values are obtained for façade regions using the SD estimation method proposed in this paper shown in Figure 3(c).

Moreover, later use of the third dimension in robust 3-D surface normals estimation provides much better results of extracting façades by rejecting non façade points.

Figure 4 presents the comparison of the façade extraction results obtained using the SD estimates in Figure 3. Taken the façade point candidates extracted by thresholding SD (Figure 3(a) and Figure 3(c)) as inputs, 2-D morphological operations (area opening and dilation) as proposed in [42] and robust 3-D surface normals information as suggested in this paper are performed to reject false positives. The final extracted façades are shown in Figure 4(b) and Figure 4(d), respectively.
image estimated via [42]; (c) $SD$ estimated via M-estimator based directional filter proposed in this paper. Higher $SD$ regions depict probable façade points. $SD$ is colorcoded with colorbar representing points/m². (b) and (c) share the same colorbar. Note $SD$ estimated in (b) is rastered image obtained by projecting all points onto $xy$ grid as compared to (c) where $SD$ is directly computed for each point.

Figure 4: Extraction of façade regions/points using $SD$ estimation results from Figure 3: (a) The building façades obtained by thresholding the rastered $SD$ image of Figure 3(b) by $TH$ value; (b) the final extracted façade after 2-D morphological operation on (a) as proposed in [42]; (c) the TomoSAR points whose $SD$ estimated in Figure 3(c) are above $TH$; (d) The extracted façades from (c) by utilizing robust 3-D surface normals information. The threshold value $TH$ used here is the 2 points/m² (empirically found to be optimum in [42]).

C. Segmentation of individual façades

The extracted façade points belong to different façades. Clustering of points belonging to the same façade is therefore needed. Firstly, a coarse clustering is performed using density based clustering algorithm [43]. It involves the notion of density connectivity between the points. For example two points are directly density connected to each other if one is in the neighborhood vicinity of the other point. If the two points are not directly connected to each other, still they can be density connected to each other if there is a chain of points between them such that they all are directly density connected to each other. Two parameters that control the clustering process include the neighborhood parameter $\varepsilon$, i.e., the radius in case of sphere or cylindrical neighborhood, and the minimum number of points $\text{MinPts}$ in the $\varepsilon$-neighborhood for any particular point. The resulting clusters $K_i$ thus contains points such that all the points in any particular cluster are density connected to each other but are not density connected to any other point belonging to another cluster. Moreover, each point inside any particular cluster $K_i$
belongs to one of the three categories (Figure 5) [43]:

- Core points: A point is labeled core point if it contains, within its \( \varepsilon \)-neighborhood, MinPts number of points;
- Border points: A point is considered border point if it is within \( \varepsilon \)-neighborhood of any core point but itself is not a core point and does not have MinPts neighbors;
- Outliers: A point neither core point nor border point is termed as an outliers i.e., any point which do not have density (points) greater than MinPts within its \( \varepsilon \)-neighborhood and also is not the neighbor of any other point;

Density connected clusters containing only core and boundary points are used for further processing.

Figure 5: Density based clustering. Points a and b are directly density connected to each other whereas points a and c are density connected to each other since there is a chain of points between them such that they all are directly density connected to each other.

The above process however may merge points of two or more adjacent façade segments into a single cluster. To reconstruct individual façade segments, separation of these segments is therefore necessary. It is done by mapping the façade points in Gaussian image and then employing mean shift clustering.

Let us assume that a coarsely clustered segment \( K \), consist of one or more vertical adjacent façades \( F_j, j = 1, \ldots J \). An image of a map \( M: F \rightarrow F^2 \) that assigns each point in \( F \) to its respective unit surface normal is known as Gaussian image \( GI \) of \( F \) [65]. Flat \( F \) (i.e., planar surface) should ideally be represented by a point in \( GI \). Figure 6 conceptually illustrates this in an ideal scenario. In practice, surface normals are estimated locally and may fluctuate from one point to another as practical data often
contains errors in 3-D positions. But, if the estimation of normals is robust enough, a surface mapped to GI should be represented as a dense cluster of points in GI. The shape of clusters in GI corresponds to the geometry of connected surfaces [44]. The number of clusters in GI tells the number of surfaces in the spatial domain.

Figure 6: Gaussian image of three connected planar surfaces: (a) Arrows indicate surface normal vectors \((n_{\text{red}}, n_{\text{green}}, n_{\text{blue}})\) to the respective surfaces; (b) All points belonging to one particular surface are mapped to same identical point in GI (ideal scenario).

If we assume \(p_r = 1, \ldots, m\) to be 3-D points and \(n_r\) as their corresponding 3-D unit normal vectors belonging to one of the coarsely clustered segments \(K_i\), then the density at any normal point \(n_q(q \in r)\) in GI (feature space) is defined as [44]:

\[
D_{n_q} = \frac{c}{mb^3} \sum_{r=1}^{m} g \left( \frac{n_q - n_r}{b} \right) \tag{5}
\]

where \(b\) is the bandwidth parameter and \(g(x)\) is a nonnegative, non-increasing, piecewise continuous function with definite integral i.e., \(\int_0^\infty g(x)dx < \infty\). From the concept of kernels [45], the function \(g(x)\) is defined as the profile of the radially symmetric kernel \(G(x)\) satisfying \(G(x) = c \|x\|^2\) where \(c\) is a normalization constant ensuring that \(G(x)\) integrates to 1.
kernels, such as the unit flat kernel and the Gaussian kernel can be used to define the density $D_{n_q}$. However, the latter with the profile function $\exp\left(-\frac{\|n_q - n_r\|^2}{b^2}\right)$ has been used in this work.

Density $D_{n_q}$ is higher for points that belong to planar or parabolic surfaces and lower for points that lie at the transition edges between the surfaces [44]. These higher density points in the GI are identified and clustered using meanshift (MS) clustering procedure. MS is a mode seeking procedure and works iteratively by shifting every data point towards the weighted mean of points within its neighborhood. The shift vector $m(n_q)$ always points towards the direction of the maximum increase in the density $D_{n_q}$ [66] and is computed as:

$$m(n_q) = \frac{\sum_{r=1}^{m} n_r \exp\left(-\frac{\|n_q - n_r\|^2}{b^2}\right)}{\sum_{r=1}^{m} \exp\left(-\frac{\|n_q - n_r\|^2}{b^2}\right)} - n_q$$  

Applying MS in GI produce clusters whose corresponding points in spatial domain represent different façades. However, it is also possible that spatial points corresponding to any one particular normal cluster in GI may belong to two or more different façades. This can happen if points of two or more façades that are "nearly" parallel to each other (i.e., having close normal directions) are present in $K_i$. To resolve this, density based clustering is again performed in the resulting clusters for spatial separation of parallel façades points clustered into one group. Finally, clusters with very few points are removed from further processing for robust reconstruction.

D. Cluster identification

Each cluster is further classified into flat or curved surface by analyzing derivatives of the local orientation angle $\theta$. $\theta$ for each 3-D point is equal to the azimuthal angle of the corresponding computed surface normal:

$$\theta = \arctan\left(\frac{\lambda_{3y}}{\lambda_{3x}}\right)$$  

(7)
Where $\lambda_{3x}$ and $\lambda_{3y}$ represents the $x$ and $y$ components of the surface normal $\lambda_3$ of any 3-D point. Ideally, the flat surfaces should have constant orientations, i.e., zero derivatives compared to the curved surfaces that have gradually changing orientations (see Figure 7). We exploit this fact and compute the first derivative $\theta'$ of the orientation angle for each façade footprint. Since the original orientation derivatives $\theta'$ are usually noisy, all the points are first projected to the first principal axis and polynomial fitting is later applied for denoising. Based on the behavior of $\theta'$, façade footprints are classified as flat or curved.

![Figure 7: Illustration of orientation angle for flat and curved vertical footprints (top view); (a) Arrows indicate pattern of change in orientation (azimuthal) angles of ten points on each vertical surface; (b) plots their respective orientation angles.](image)

**E. Modeling of façades**

Identified façade clusters in $xy$ plane are then modeled using the following general polynomial equation [42]:

$$f_p(x, y) = \sum_{q=i}^{p} a_q x^i y^j \quad i + j \leq q$$  \hspace{1cm} (8)

where $i$ and $j$ are permuted accordingly, $p$ is the order of polynomial, the number of terms in the above polynomial is equal to $(p + 1)(p + 2)/2$. Cross terms are introduced in the model in case of rotated local coordinate system. To solve (8), we restrict ourselves to 1st and 2nd order (i.e., flat with $\max(i, j)=1$ & curved with $\max(i, j)=2$). The coefficients $a_q$ are estimated using weighted total least squares (WTLS) method where total least squares is utilized to cope for localization errors of TomoSAR points.
in both $xy$ directions and the weight of each point is assigned equal to its corresponding $SD$. The weighted polynomial fitting (residual) error $f_{err}$ is minimum for the case where we have unrotated local coordinate system reducing right hand side of (8) to $\sum_{i=0}^{p} a_i x^i$ (i.e. with no cross terms). In case of rotated local coordinate system (which is often the case), we perform following steps to obtain consistent parameter estimates of all façades in a global coordinate system:

- Rotate the points by rotation angle $\omega$ and compute polynomial fitting error $f_{err}$ by applying WTLS method;
- Consider coefficients computed with $\omega_{min}$ that gives the minimum polynomial fitting error $f_{err}$ as polynomial terms depicting unrotated points in the global coordinate system. $\omega_{min}$ is computed by using an unconstrained nonlinear optimization procedure to find the minimum of the error function $f_{err}$ by varying $\omega$ over 0–360 degree range via Nelder-Mead simplex algorithm [67];
- Rotate the computed polynomial by replacing the unrotated $(x, y)$ axis terms by their rotation counterparts $(x \cos \omega + y \sin \omega, -x \sin \omega + y \cos \omega)$ to yield polynomial terms $a_q$ in global coordinates.

**F. Removing conflicting segments**

After estimation of model parameters, the next step is to describe the overall shape of the building footprint by further identifying adjacent façades pairs and determining the intersection of the façade surfaces. The adjacency of façades is usually described by an adjacency matrix $AM$ that is built up via connectivity analysis [42][21]. Identified adjacent façade segments are used to determine the vertex points (i.e., façade intersection lines in 3-D) by computing the intersection points between any adjacent façade pair.

Determination of these intersection points can sometimes become difficult if the transition points are segmented as isolated small clusters (also referred to as conflicting segments) rather than part of the corresponding adjacent façade segments. As a consequence, it gets complicated to find a legitimate
adjacent façade pair from which intersection points should be computed. To resolve this issue, conflicting segments must be removed prior to vertex points computation. To illustrate how they are removed in an automatic manner, an example is shown in Figure 8. The labeled line segments indicate the reconstructed façade segments of two different buildings A and B. Endpoints of each segment are denoted as "Δ" and "·". AM represents the built adjacency matrix where "1" and "0" denotes the adjacent and not-adjacent conditions respectively. Among the labeled segments, segments [7 4 1 3 6] are “valid” façades while segments [2 5 8] are the conflicting segments.

Figure 8: Example illustrating the removal of conflicting segments

Following steps are performed for automatic removal of these conflicting segments:

- The connected series matrix ConnSeg is determined from AM such that rows of ConnSeg represent a set of distinct series of adjacently connected segments, e.g., the i'th row of ConnSeg $Seg_i = \{s_j | j = 1, ..., n\}$ represents n segments (i.e., $s_1$-$s_n$) that are adjacently connected. In Figure 8, since there exists only two series of adjacently connected façade segments (i.e., belonging to two buildings), ConnSeg therefore consists of two rows only where the first row contains façade segments $Seg_1 = [2 4 5 7]$ while the second row comprises of segments $Seg_2 = [1 3 6 8]$.

- For each segment, the largest segment that is connected to each endpoint can be identified. Their indices are recorded in a two column matrix E that captures such an “endpoint” - “largestsegment” relationship. E.g. the "·" endpoint of segment 1 shown in Figure 8 is connected to two segments 8 and 3. Since segment 3 has a larger length than segment 8, therefore 3 is assigned to this endpoint of segment 1 in E. The endpoint matrix E for both buildings is depicted on the right side of Figure 8. Zeros in E represent the condition of no adjacent façade at that endpoint.
Applying union operation to all elements in E results in a matrix RetainSeg whose elements contain all building façades that should be retained. Conflicting façades, i.e. the ones that are not part of RetainSeg are removed. For the example shown in Figure 8, the union of elements in E gives the RetainSeg [1 3 4 6 7] (zeros are not considered). Subsequently, the segments that are not part of RetainSeg, namely [2 5 8], are removed.

Pseudo code for the above procedure is given in the Table 1.

Table 1: Procedure to remove conflicting segments

| Given: Endpoints matrix EndPts containing endpoints of the reconstructed façade segments & Adjacency matrix AM = \{a_{vw} | v, w = 1, \ldots, k\} such that a_{vw} = 1 & a_{wv} = 1 only for conditions where façade Segments v and w are adjacent to each other, otherwise a_{vw} = 0. k is the total number of façades reconstructed. |  |
|---|---|
| 1. Initialize: RemoveSeg := Ø |  |
| 2. for i = 1 to number of rows in ConnSeg |  |
| 3. Select the adjacently connected series of segments Seg_i = ConnSeg(i,:) |  |
| 4. if number of elements in Seg_i > 2 |  |
| 5. for j = 1 to n (number of segments in Seg_i) |  |
| 6. Extract adjacent segments of s_j in Seg_i from AM. |  |
| 7. Extract endpoints of all the segments in Seg_i from EndPts. |  |
| 8. Compute the Euclidean distance of both endpoints in s_j from end points of all other segments s_k (j ≠ k) in Seg_i. |  |
| 9. Find out the largest adjacent segment l_1 connected to one endpoint. Similarly find the largest adjacent segment l_2 connected to the other endpoint of s_j. |  |
| 10. Store l_1 and l_2 in a two column matrix E(j, 1:2). In case of no adjacent segment to any of the endpoints, insert 0 in the corresponding position in E. |  |
| 11. end for |  |
| 12. Separately apply union operation to both column of E and store the unique result after excluding 0s in the row matrix RetainSeg. |  |
| 13. Compare Seg_i and RetainSeg and insert segments of Seg_i that are not members of RetainSeg in the matrix RemoveSeg(i,:) for removal. |  |
| 14. end if |  |
After removing conflicting segments, the vertex points are computed from the intersection of valid adjacent segments to complete the reconstruction process.

G. Refining reconstructed façades

Sometimes the reconstructed façades remain either incomplete or are broken into more than one segment due to the following reasons: 1) Higher building structures present nearby can partly (or fully) occlude the façades of lower buildings; 2) Due to the geometrical shape, only very few points are available at some parts of building façades. In order to overcome this issue, in this section we propose a procedure that tries to refine the reconstructed façades by inserting additional segments between the broken regions and extend those façades that remain incomplete.

Vertex points computed from the previous section are separated into two types: First type consists of vertices that are computed from the intersection of two adjacent façades, while the second type consists of the other vertices representing “open” endpoints. For each series Seg, made up of \( n \) segments (i.e., \( i \)th updated row of ConnSeg matrix after removing conflicting segments), there exists two open vertices and \( \binom{n-1}{2} \) intersection vertices. Refinement operations including insertion of additional segment to connect broken façades and extension of incomplete façades are carried out only on the second type i.e., open endpoint vertices.

If we denote an open end vertex as \( v_o \), and an intersection vertex as \( v_i \) (see Figure 9(a)), then the refinement procedure for any one particular adjacently connected series Seg, having two open vertices is described in the following Table 2:

<table>
<thead>
<tr>
<th>Table 2: Refinement procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> Any one particular adjacently connected series Seg,</td>
</tr>
<tr>
<td>1. Select one of the two open vertex point ( v_o ),</td>
</tr>
<tr>
<td>2. Find the nearest open vertex point ( \hat{v}_o ) from ( v_o ) that belongs to another adjacently connected series</td>
</tr>
</tbody>
</table>
\( \text{Seg}_i (j \neq i) \) and compute their midpoint \( \text{mid}_{ij} \).

3. Locally compute maximum height value \( h_{\text{max}} \) and height standard deviation \( h_{\sigma} \) of 3-D points within neighborhood vicinity of \( v_o, \hat{v}_o \) and \( \text{mid}_{ij} \). Additionally, compute local orientation \( \theta \) at \( v_o \) and \( \hat{v}_o \).

4. Check following three conditions (C1, C2, C3):
   
   \[ \| \hat{v}_o - v_o \| < 2\varepsilon \]
   
   \[ \text{abs}(h_{\text{max} \text{ at } v_o} - h_{\text{max} \text{ at } \text{mid}_{ij}}) < T_{\varepsilon} \text{ & abs}(h_{\text{max} \text{ at } \hat{v}_o} - h_{\text{max} \text{ at } \text{mid}_{ij}}) < T_{\varepsilon} \]
   
   \[ \text{abs}(\theta_{v_o} - \theta_{\hat{v}_o}) > 45^\circ \]
   
   If conditions C1 & C2 are true, then check condition C3.
   
   If C3 is also true
   
   Insert two additional segments with vertices \( (v_o, p_{v_o \hat{v}_o}) \) and \( (\hat{v}_o, p_{\hat{v}_o \hat{v}_o}) \) and then go to step 1 (\( p_{v_o \hat{v}_o} \) is the point of intersection between two segments).
   
   else
   
   Insert an additional segment with vertices \( v_o \) and \( \hat{v}_o \) and go to step 1
   
   end if
   
   else
   
   proceed to the next step 5.
   
   end if

5. Determine point \( v_p \) at a distance \( 2r_{\text{oc}} \) away from \( v_o \) using the estimated model parameters (\( r_{\text{oc}} \) is the radius used for defining local neighborhood around \( v_o \))

6. Similarly as step 3, locally compute \( h_{\text{max}} \) and \( h_{\sigma} \) of 3-D points within neighborhood vicinity of \( v_p \).

7. Check following two conditions (C4, C5)
   
   \[ \text{abs}(h_{\text{max} \text{ at } v_p} - h_{\text{max} \text{ at } v_o}) < T_{\varepsilon} \]
   
   \[ \text{abs}(h_{\sigma \text{ at } v_p} - h_{\sigma \text{ at } v_o}) < T_{\sigma} \]
   
   if both C4 and C5 are true, then recursively extend the vertex \( v_o \) towards \( v_p \) (see Figure 9(c)) and finally add a new segment with vertices \( (v_o, v_p) \)

8. Continue steps 1 to 7 for the second open vertex point of the series \( \text{Seg}_i \).
In Table 2, steps 2-4 tries to cope with the broken façades, while steps 5-7 deals with the incomplete façades. Conditions $C_1$ and $C_2$ in step 4 imply that the two segments are considered part of the same (broken) building façade if both segments are not far enough from each other and at the same time possess data points in between that have close maximum height values. $h_{\text{max}}$ is taken as the mean of at least ten maximum height values (i.e., if there are less than ten points available then $h_{\text{max}}$ is taken as mean of all those point). If conditions $C_1$ and $C_2$ in step 4 are met, the algorithm then checks the condition $C_3$. If the two segments belong to the same façade, a segment with vertices $(v_o, \hat{v}_o)$ is inserted that fills the empty (i.e., broken) regions of the façade. On contrary, if the open vertex pair $v_o, \hat{v}_o$ are not part of the same façade but rather belong to two different façade segments (determined via difference in the local orientation angle $> 45^\circ$), then point of intersection $p_{v_o,\hat{v}_o}$ is computed and instead of inserting one segment, two segments with vertices $(v_o, p_{v_o,\hat{v}_o})$ and $(\hat{v}_o, p_{v_o,\hat{v}_o})$ are inserted. Figure 9(b) graphically depicts such a situation where grey open vertices of segment 4 and 5 are (assumed to be) within $2\varepsilon$ distance but have difference in the local orientation angle of $90^\circ$. The grey dotted line shows the addition of new segment without checking condition $C_3$. When $C_3$ is taken into account, two segments shown in black dotted line are inserted.

In contrast, if any of the conditions $C_1$ or $C_2$ fails, then the algorithm tries to extend the open vertex point $v_o$ by imposing constraints $C_4$ and $C_5$ present in step 7. Similar to $C_2$, the condition $C_4$ ensures that the extended point have the closer maximum height value. The condition $C_5$ ensures that the local 3-D points have certain standard deviation. It is necessary to make sure that the extension is not carried out in the direction that deviates from the façade footprint. I.e., it avoids the extension if the local 3-D points around $v_p$ belongs to other non-façade objects e.g., roofs etc. The problem is illustrated in Figure 9(d) where the grey open vertex can potentially extend over the roof region if the condition $C_5$ in step 7 is ignored. Thus imposing this constraint helps in limiting this false extension.
Figure 9: Vertices for refinement. Grey rectangles depict the 2-D building footprint from the top: (a) shows the total of 5 vertices out of which 4 are open endpoint vertices and one is intersection vertex computed from the intersection of segments 1 and 2; (b) depicts the situation where ignoring condition $C_3$ would yield false segment addition shown as grey dotted line. The grey arrows indicate the local orientation angle $\theta$ at open vertices $v_o$ and $\hat{v}_o$. Two black dotted lines represent the two correct inserted segments between $v_o$ and $\hat{v}_o$; (c) depicts the recursive extension procedure of the open vertex $v_i$. $v_p$ represent the intermediate extension points where as the $v_p'$ denotes the final point; (d) illustrates the direction of extension of grey open vertex over the roof region. This can happen if we only consider the $h_{\max}$ and ignore the local standard deviation of height $h_o$.

Finally, the computed vertex points (i.e., the intersection vertices and the open vertices before and after refinement) along with their estimated model parameters are used to reconstruct the 3-D model of the building façades.

**IV. EXPERIMENTAL RESULTS & VALIDATION**

**A. Data set**

To validate our approach, we tested the algorithm on TomoSAR point clouds generated from a stack of 25 TerraSAR-X high spotlight images from ascending orbit only using the Tomo-GENESIS software developed at the German Aerospace Center [68]. The test area covers approx. 2 km² in the high rise part of the city Las Vegas. The number of TomoSAR points in the area of interest is about 1.2 million. Figure 10(a) shows the optical image of our test area while Figure 10(b) shows the corresponding TomoSAR point cloud in universal transverse mercator (UTM) coordinates.
Figure 10: Dataset: (a) Optical image of the test area in Las Vegas © Google; (b) TomoSAR points in UTM coordinates of the corresponding test image. Height of TomoSAR points is color-coded [unit: meter].

B. Results - Extraction of façade points

The result of applying $SD$ estimation procedure is illustrated in Figure 11(a). The two parameters $r$ (radius of the neighborhood cylinder) and $d$ are empirically set to 5m and 0.9m respectively according to the point density of the data set. One can observe that $TH$ value influences the number of extracted façade points. Lower $TH$ value results in higher completeness but lower correctness. In [42], we showed the results of estimating $SD$ with varying area sizes and found that a kernel window of size 3x3 m$^2$ and threshold $TH$ value of about 2 pts/m$^2$ results in best trade-off in terms of completeness and correctness with this class of data. 2 pts/m$^2$ works well for high rise buildings but might ignore relatively smaller façades. Therefore to extract lower façades (and also to automate the procedure), we set the $TH$ to the maximum of $SD$ histogram value. This, as described in section 3, includes not only the façade points but additionally also some non-façade points with relative high $SD$, e.g., roof points. To reject these points from the set of extracted points after $SD$ thresholding, surface normals information is utilized. Figure 11(b) shows the extracted façade points by retaining only those points having normals between $\pm 15$ degrees from the horizontal axis (or equivalently $\pm 90$ degrees from the vertical axis).
Figure 11: Façade points extraction: (a) Scatterer (point) density with radius $r = 5m$ and inliers $d = 0.9m$; (b) Extracted building façade points. Colobar indicates $SD$ and height in meters in (a) and (b), respectively.

C. Results - Automatic clustering of extracted façade points

Once the façade points are extracted out, the next step is to cluster them into segments where each segment corresponds to an individual façade. For this, we apply the clustering procedure using the cylindrical neighborhood definition and cluster all the points with parameter settings: $\varepsilon = r = 5m$ and $\text{MinPts} = 2$. Here an important point to notice is that two buildings are considered distinct only in a case when points belonging to façades of two different buildings are separated by $\varepsilon$. Setting $\varepsilon$ too small can cause points belonging to single cluster (i.e., corresponding to an individual façade) to break into more than one cluster. On the other hand, larger values of $\varepsilon$ tend to merge points of the nearby façades into one cluster. The value of $\varepsilon$ is therefore empirically chosen according to the length and distance among the buildings in the area of interest and implicitly indicates the assumption that two individual façades belong to different buildings are farther apart than $5m$ radius.

Setting parameter $\text{MinPts}$ equal to 2 implies that points are connected to one cluster even if there is a single neighboring point among them. This parameter helps in removing outliers that do not have any neighboring point and produce clusters similar to the clusters obtained from dendogram cut at $\varepsilon$ in case of hierarchical clustering using single link metric [43]. Increasing $\text{MinPts}$ can help in retaining more stable core points but on the other hand can also break the clusters into two or more clusters. This property is sometimes useful in cases when different clusters are merged together by a thin line of
points. Estimating the exact value of MinPts is however very much dependent on the dataset and certain heuristics based on the "thinnest" cluster in the dataset, e.g., k-distance graph can be employed [43].

In order to reconstruct individual façades, these density based coarse clusters need to be further clustered. To this end, mean shift clustering has been applied to the coarsely clustered segments in their normal feature space (in GI domain). Figure 12(b) shows the estimated orientation angle \( \theta \) for extracted façade points from single building shown in Figure 12(a). The variation in orientation angle is quite evident and allows mean shift to cluster points having similar orientations together. Further separation of points in the spatial domain is also required in some cases where the spatially separated points are clustered into one segment. This happens when these points belonging to different façades have similar normals and are spatially closer. Density based clustering is therefore again applied for spatial separation of the clusters within clusters.

![Image](image.png)

Figure 12: Fine clustering results after applying mean shift clustering using Gaussian kernel with bandwidth \( b = 0.4 \) to the coarsely clustered segments in their normal feature space (in GI domain): (a) TomoSAR points of one particular density connected cluster (top view). Colorbar indicates height in meters; (b) Corresponding orientation angle in degrees; (c) Non clustered (top) and clustered (bottom) points in the Gaussian image of points in (a); (d) Resulting clustered points in 3-D.

D. Results - Reconstructing façades

Prior to reconstruction, the segmented façades are first classified to flat and curved surfaces by analyzing derivatives of the local orientation angle \( \theta \). A slope value of 0.3 (\( \approx 17 \) degrees) is set by empirically testing the computed orientation angles of all the buildings in the area of interest to distinguish flat and curved surfaces.
After identification, appropriate model parameters are estimated from the core points of the individual clusters. Vertex points are then determined by computing intersections of the adjacent segment pairs. However in doing so, smaller clusters occurring at façade transition regions behave as noisy segments in the reconstruction procedure. A practical example of these so-called conflicting segments is shown in Figure 13 (a). Following the procedure explained in Table 1 (see Section IIIF), the conflicting (reconstructed) segments occurring at the transitional regions of individual buildings are removed prior to the vertex point computation as exemplified in Figure 13 (b).

Once these transitional clusters are removed, the intersection vertices are determined by computing intersection point of the two adjacent façades. Refinement operation is then carried out on the open vertices to insert additional segments between the broken façade regions followed by extension of incomplete reconstructed façades.

Figure 14(a) and (b) depicts the reconstructed façades models of the area of interest before and after refinement, respectively. Green lines shows reconstructed façade footprint before refinement. Blue lines indicate additional segments that are added between the vertices of those broken façades that meet the conditions present in step 4 in section IIIG while the red lines are subsequent extensions of the open
vertices after filling the break regions.

E. Results - Validation

The actual ground truth data is missing for exact qualitative evaluation of the approach. In order to provide some quantitative measures of the algorithm performance, we manually counted the actual number of façades that were to be reconstructed. Total of 141 façades are present in the dataset out of which 7 are curved façades and remaining 134 are flat. Prior to refinement operation, the algorithm reconstructed in total of 176 façades, i.e., higher than the actual façades present in the dataset. As already stated in section IIIG, this is because some individual façades have been broken down into two or more segments due to discontinuity in the number of points available in the dataset. After refinement, 29 insertion segments (27 single and 2 double based on the condition C3 in Table 2) are added between the broken façade regions where as 43 façades have been extended. In the final reconstruction, we obtain 147 reconstructed façades. I.e., all 141 façades are successfully reconstructed; among them 5 façades remain broken (counted as additional 5 façades) and there is one case of false alarm which will be explained later. Besides the 5 cases, we also find 7 façades that are not extended and therefore remain incomplete. This is however due to the inadequate number of points available in the data.
Figure 14: Reconstructed façades: (a) shows the 2-D view of the façade footprints overlaid onto the optical image prior to refinement; (b) shows the 2-D view of the façade footprints overlaid onto the optical image after refining with parameters settings $T_s = 5m$ and $T_p = 2.5m$. 
As mentioned earlier, there is also one case, shown in Figure 15, which is considered as false positive (i.e., a façade not actually present but reconstructed by the algorithm). As can be seen in Figure 15(c), the reconstructed segment is actually a bridge for pedestrian crossing. Higher number of scatterers is retrieved over the bridge due to its apparently metallic structure. Moreover, the bridge is also covered from the top and therefore scatterers are obtained on top and bottom and as well on the metallic rods connecting the upper and lower surfaces of the bridge. The estimated surface normal of these scatterers thus give higher horizontal component and as a consequent these scatterers are wrongly classified as façade points by satisfying both extraction constraints: higher $SD$ and higher horizontal component of the surface normals.

It is also interesting here to mention that in Figure 14 some small vertical structures on roofs of the buildings or on ground are very well-reconstructed. Figure 16 shows some examples of such objects that might visually appear (or interpreted) as false reconstructions in Figure 14 but are actually vertical structures (e.g., advertisement boards, monuments etc.).
Figure 16: Reconstructed façades on left. Their 3-D view on the right (© Google Earth).

Figure 17: 3-D view of the final façade reconstruction. The axis is in meters range and has been translated to the origin for better metric clarity by subtracting UTM easting and northing values by their respective minimum values present in the reconstructed vertices.

Finally, in Figure 17, we present the final reconstructed façades in 3-D. As depicted in [42], the shown reconstructed façade model can be used to refine the elevation estimates of the raw TomoSAR points. Moreover, with known deformation estimates of the scatterers, such a model can also lead to the
reconstruction of dynamic city models that could potentially be used to monitor and visualize the dynamics of urban infrastructure in very high level of details.

V. Outlook & Conclusions

In this paper we presented an automatic (parametric) approach for robust façade reconstruction for large areas using TomoSAR point clouds. The approach is modular and works directly on unstructured 3-D points. It allows for a robust reconstruction of both higher façades and lower height structures, and hence is well suited for urban monitoring of larger areas from space. A few points, however, need be addressed:

- During SD estimation, the continuity of an individual façade can be broken due to limited number of available points. This may result into two or more segments of the same façade. We attempted to cope with this problem by refining the reconstructed façade footprints via insertion and extension operations. Still, the lack of measurements prevents the complete resolution of this problem. Use of 2-D ground plans or cadastral maps can be helpful in this case.

- Since the satellite orbits are bound to pass close to the poles of Earth, we may fail to reconstruct building façades facing North or South due to the missing of measurements. One way to rectify this is by using fused point clouds (i.e., both ascending and descending) and/or inserting new segments by simply connecting the endpoints of the missing façades if they match a certain criteria to get the complete shape of the building footprint.

- The presented approach is much better option to detect shape of the building when dense points on the façades are available. However, in cases (usually for lower height buildings) when no or few façade points are available, one can try to extract roof points and reconstruct the 2-D footprint. This could help in resolving the problems related to the visibility of façades mainly pointing towards the azimuth direction.

In the future, we will work over these considerations and will extend the algorithm towards object based TomoSAR point clouds fusion and automatic building roof reconstruction.
ACKNOWLEDGEMENT

This work is partially supported by the Helmholtz Association under the framework of the Young Investigators Group “SiPEO”. This work is also part of the project “6.08 4-D City” funded by the IGSSE of Technische Universität München and the German Research Foundation (DFG, Förderkennzeichen BA2033/3-1).

REFERENCES


