Detection of Defective Regions in 3D Reconstruction to Support Image Acquisition

Seiya Ito *, Naoshi Kaneko *, Takeshi Yoshida **, and Kazuhiko Sumi **

Abstract: To successfully reconstruct a three-dimensional (3D) model from images, it is necessary to make the user aware of whether sufficient data has been obtained. We propose a novel approach to detect defective regions in 3D reconstruction during image acquisition. Our method uses line-based segmentation to segment the acquired images into structural and non-structural regions. Then, using the 3D structures derived from these images, defects detected in the structural region are used to suppress spurious artifacts. Visualization of a defective region in a newly acquired image allows the user to comprehend the reconstruction state and adjust image acquisition. The proposed method was experimentally demonstrated to work successfully in a range of outdoor environments.

Key Words: UAV, image acquisition, 3D reconstruction, defect detection.

1. Introduction

Three-dimensional (3D) reconstruction can be used to represent terrain and building layouts from data such as multi-view stereo images or laser scanning. Recently, aerial photography by unmanned aerial vehicles (UAVs) has become widespread because they can shoot high-resolution images from multiple perspectives which may be inaccessible when using ground-based cameras or cameras positioned from high altitude. Since it is possible to reconstruct dense 3D models, the value of image-based 3D reconstruction has increased. In recent years, the demand for 3D reconstruction of buildings, for the purpose of construction and/or reconstruction of dilapidated structures, has increased.

In general, image-based 3D reconstruction requires the use of multiple images, taken from multiple perspectives. The 3D model is reconstructed from feature points observed in the images and represented as 3D points. Since the 3D model depends on the feature points from the images, it often contains multiple defects due to a failure of feature matching, resulting from occlusion, illumination condition, or uniform texture. For instance, Fig. 1 shows that the 3D model from the UAV images has defective regions due to lack of features resulting from uniform texture. In such cases, additional images must be acquired to provide the missing features. However, this post hoc image acquisition is often ineffective for two reasons: the scene changes over time or the acquisition process itself is too costly and time consuming. We believe that the detection of defects (within the 3D model during image acquisition) reduces the frequency of shooting by acquiring images suitable for 3D reconstruction.

2. Related Work

Structure from motion (SfM) is a technique for reconstructing the 3D structure of a scene using the images from different viewpoints. The SfM has become very popular for publicly available software like Bundler [2],[3]. The development of SfM enabled the application to large-scale environments and
unordered image sets [4],[5]. The SfM strategies which have been proposed recently include incremental [6], hierarchical [7], and global approaches [8]. All approaches commonly start with feature extraction and matching. The resulting scene graph provides the basis for the reconstruction process. The SfM initializes the 3D model with two carefully selected images. Then, it incrementally registers new images, triangulates scene points, filters outliers, and refines the reconstruction using bundle adjustment. These approaches assume that all images are acquired in advance, and it is impossible to confirm the reconstruction before the geometric computation is complete. Although incremental SfM can confirm the reconstructed model during the reconstruction process, images taken from the UAV may not be usable as inputs.

In recent years, simultaneous localization and mapping (SLAM) approaches have been proposed in which the 3D model is constructed from image sequences taken within a short time [9],[10]. Although it is possible for the user to confirm the reconstruction model on site, these approaches are often limited in terms of the scene size, or require high framerate image sequences to ensure accurate camera positioning. An alternative direct SLAM method has been proposed, in which sparse depth maps are recovered, based on epipolar line scanning [11]. While this is effective in large-scale environments, however, it is unsuitable for use with high-resolution image sequences since it requires depth estimates to be available for many pixels.

To recover a 3D model from high-resolution still images, online SfM approaches have been proposed in which a 3D model is updated as soon as new images are acquired [12],[13]. Hoppe et al. [12] used a model built from an online SfM process to visualize the reconstruction states. The 3D model was represented by 3D points, and a surface model was computed from those points. The reconstruction states, including the number of image overlaps between cameras and the accuracy based on the density of 3D points, were indicated by a heat map on the surface model. This research makes it possible to track the reconstruction states and directly understand the image acquisition process. However, the problem is that defects on the surface may deceive the user into believing that sufficient data is being acquired, when in fact it is inadequate. In addition, Hoppe’s approach requires the user’s intervention to analyze the reconstruction states.

Our method makes it possible to suppress spurious artifacts, appearing in non-structural regions such as the sky or the ground surface. By applying line-based image segmentation to extract such structures, our method automatically detects defective regions without requiring user input.

3. Proposed Method

3.1 Overview

The proposed method is based on detecting defective regions within the 3D model during the image acquisition process. Figure 2 shows the workflow of our method. Each time a new image is captured, it is integrated into the 3D model. In parallel, the image is segmented into superpixels to allow defects to be detected in the artificial structures. If the quality and completeness of the reconstructed model is insufficient, defect detection is performed using a segmented image. Following Hoppe et al. [12], our process evaluates the quality from the number of image overlaps and the density of 3D points. If the density of the 3D points is too low, despite the large number of image overlaps, the region is likely to constitute a defect. To detect such defective regions, two scores are calculated, using reprojected images. Our method can detect defective regions in the course of image acquisition.

3.2 Online Structure from Motion

To identify a defective region within the 3D reconstruction during image acquisition, the SfM problem must be solved online. We again follow the approach of Hoppe et al. [12]. In the initial stage, we extract and describe scale-invariant feature transform (SIFT) features [14] from a new image $I$. The next stage involves feature matching. We derive the visual appearance of image $I$ from its features using the bag of visual words (BoVW) technique [15]. Then, $N$ images are selected by the BoVW similarity scores, and feature matching is performed with $N$ images and image $I$. This approach can efficiently compute the camera position where image $I$ is acquired and the 3D points. In the third stage, camera positions and 3D points associated with the newly acquired $N$ images can be optimized by local bundle adjustment [13]. Although this allows recon-
struction of the 3D model in a constant time, the error of the 3D model increases as images are acquired. To prevent scene drift, we detect loops and verify feature correspondence of the 3D points across all images. If a loop is detected, or the error of the 3D model exceeds a threshold, the 3D points are merged and the camera positions and 3D points are optimized by global bundle adjustment [16].

3.3 Line-Based Image Segmentation

In many cases, the aim of acquiring images is to reconstruct a certain scene. If we are familiar with the target object in advance, it can be identified using image matching, but, recognition is challenging in the absence of such prior knowledge. As it is unnecessary to reconstruct irrelevant regions such as sky and ground, the target scene objects can be simply represented as planes [17] or lines [18].

To extract the target structures, we perform line-based segmentation of each acquired image (Fig. 3). This process has three steps: (1) extraction of contours from the straight-line segments obtained using the line segment detector (LSD) [19], (2) generation of superpixels using our extended simple linear iterative clustering (SLIC) algorithm [20], and (3) identification of every superpixel, that falls inside contours or contains line segments, as a region containing an artificial structure.

We use line segments as the foundation of the structure. Each line segment contains information including two endpoints and a line width. The first goal is to extract the contours of the artificial structure using the line segment information. Given an image, we obtain the line segments \( L = \{l_1, l_2, \ldots, l_m\} \), where each line segment \( l_i \) comprises two endpoints \( p_i, q_i \in \mathbb{R}^2 \). The center of each line segment \( c_i \) is simply calculated as \( \frac{p_i + q_i}{2} \). We also calculate each distance \( d_i \) from an endpoint to the center of the line segment. We set a search range \( H_i \) for each line segment \( l_i \) to the circles centered at the endpoints \( p_i, q_i \) with a radius of \( d_i \), then connect the endpoints of \( l_i \) with the overlap between \( H_i \) and \( H_j \) (Fig. 4). The exterior contours are extracted from the connected line segments.

To detect defective regions within the reconstructed model, we generate superpixels using our extension of the SLIC algorithm [20]. Several superpixel algorithms group pixels into perceptually meaningful regions which have been proposed include: graph-based method [21], Turbopixels [22], and Quick-Shift [23]. The original SLIC clusters pixels in the combined five-dimensional CIELAB color \([l a b]^T\) and image plane space \([x y]^T\) efficiently generating compact and nearly uniform superpixels. The clustering procedure begins with an initialization step where \( k \) initial cluster centers \( C_1 = \{c_1, c_2, \ldots, c_k\} \) are sampled on a regular grid. Next, each pixel is associated with the nearest cluster center. Since the expected spatial extent of a superpixel is a region with an approximate size of \( S \times S \), the distance \( D \) between a pixel \( i \) and the cluster center within a region \( 2S \times 2S \) around the superpixel center is computed. A distance measure \( D \) is defined as follows:

\[
d_c = \sqrt{(l_i - l)^2 + (a_j - a)^2 + (b_j - b)^2},
\]
\[
d_s = \sqrt{(x_j - x)^2 + (y_j - y)^2},
\]
\[
D = d_c^2 + \omega d_s^2,
\]

where \( m \) is the weighting of relative importance between color similarity and spatial proximity. Once each pixel has been associated with the nearest cluster center, an updating step adjusts the cluster centers to the mean \([l a b x y]^T\) vector of all the pixels belonging to the cluster. The assignment and the updating steps can be repeatedly made until the error between the new cluster center location and the previous cluster center location converges.

The SLIC algorithm is satisfactory in terms of boundary adherence, segmentation speed, and performance. However, there is a difference between the contours of the artificial structures extracted from the line segments and the boundaries between superpixels, and this causes unneeded regions to be selected. Following [24], we extend the SLIC algorithm to address this problem. We move the boundaries closer to the contours using a binary mask \( e \) which either represents or fails to represent the inside of the contours (i.e., the inside value is one, otherwise zero). Our distance measure \( D \) is derived as follows:

\[
d_e = \sqrt{(e_j - e)^2},
\]
\[
D_e = \sqrt{d_c^2 + \omega d_s^2},
\]

where \( \omega \) is weighting the importance between the contours and the boundaries. Figure 5 shows the comparison between the original SLIC and our extension, and demonstrates that the extended version can effectively represent the contours extracted from the line segments.

Finally, we extract an artificial region by selecting every superpixel that falls inside the extreme outer contours, or that contains line segments. This prevents a non-structural region from being classified as a defect.

3.4 Defect Detection

To confirm the quality of the reconstructed model, we use a reprojected image \( R \), which is generated by projecting the reconstructed 3D points to the camera that corresponds to the newly captured image \( I \). The most simple approach to achieve
the quality of the 3D model is to compare image I and the reprojected image R at the pixel level. However, this approach is unsuitable when using high resolution images, which have a large number of pixels. Therefore, we derive two scores to detect defective regions within the model. Let \( S = \{ S_i | i = 1, \ldots, N \} \) be a set of superpixels which obtained from the extracted artificial object region. The superpixel \( S_i \) has pixels \( P = \{ p_j | p_j \in S_i \} \). We calculate the center of gravity \( g_i \) in the superpixel \( S_i \). At the pixel \( p \) of the reprojected image \( R \), the pixel value \( v(p) \) where the reprojection point exists is one, and otherwise it is zero. The pixel \( p \) also has the number of image overlaps \( r(p) \). For each superpixel \( S_i \), we calculate the following scores:

\[
\text{Score}_{\text{r}} = \frac{\sum_{p_j \in P} v(p_j)}{|S_i|},
\]

\[
\text{Score}_{\text{d}} = \frac{\sum_{p_j \in P} d_{\text{max}} \cdot \text{dist}(g_i, p_j)}{d_{\text{max}}},
\]

in which

\[
d_{\text{max}} = \max_{p_j \in P} \text{dist}(g_i, p_j).
\]

Here, \( \text{dist}(\cdot, \cdot) \) is the Euclidean distance. \( \text{Score}_{\text{r}} \) represents the area ratio between the reprojected points and a superpixel. In deriving \( \text{Score}_{\text{r}} \), we assume that the points near the boundary of a superpixel are less important than those of the center of a superpixel. The reprojected points are derived from SIFT \[14\].

The SIFT algorithm detects feature points by calculating the extreme values of difference of Gaussian. Therefore, the feature points appear around a large color change. On the other hand, superpixels are generated by our extension of the SLIC algorithm in Eq. (5). The resulting superpixels are grouped with similar colors and by close range, and the boundaries between superpixels are along the line structure. When no line structure is observed around the pixel of interest, a boundary between superpixels is formed by color change. Therefore, the reprojected points often appear on the boundaries of superpixels. This may cause a hole in the center of the superpixel, and it becomes a defect. In \( \text{Score}_{\text{d}} \), the closer the point is to the center, the higher the score. In addition, we use \( r(\cdot) \) as the weight because it is known that reprojected points with a high level of image overlap are likely to be accurate.

However, the 3D points obtained in an online SfM process are much sparser than the 3D points reconstructed by basic SfM or a multi-view stereo algorithm such as patch-based multi-view stereo software (PMVS) \[25\]. This leads to a false detection of defective regions, even when reconstruction is successful. To avoid this, we represent the quasi-dense reconstructed 3D points as expanding reprojected points by dilation operation. Here attention must be paid to the appearance of outliers in the 3D points. These can arise from a failure of feature matching, even though the online SfM process is designed to be robust against outliers. Furthermore, the same image point can be used in the reconstruction of more than one 3D point. To prevent false detection arising from such outliers, we use three structuring elements in dilation operation, defined as follows:

\[
\begin{align*}
&7 \times 7 \quad (r(p) \geq 0.8N \land th_{\text{close}} < d_r < th_{\text{far}}) \\
&5 \times 5 \quad (r(p) \geq 0.5N \land d_r < th_{\text{far}}) \\
&3 \times 3 \quad \text{(otherwise)}
\end{align*}
\]

where \( N \) is the number of images selected in the feature matching process, \( d_r \) is the distance between camera and a 3D point, and both \( th_{\text{close}} \) and \( th_{\text{far}} \) are constant parameters. We also prevent the excessive expansion of reprojected points which are close to or far from the camera. The reprojected image \( R \) after this dilation operation can be used in the same way as the image obtained by reprojecting the dense 3D points.

The reprojected image \( R \) can be used to more clearly distinguish between a successfully reconstructed region and a defective region than the reprojected image \( R \). More precisely, since one reprojected point corresponds to one pixel, the area applied to image \( R \) is equal to the number of reprojected points in \( S_i \). In contrast, the image \( R \) derives its area according to the number of image overlaps because one reprojected point corresponds to more than one pixel.

Using image \( R \), we calculate \( \text{Score}_{\text{r}} \) and \( \text{Score}_{\text{d}} \). If \( \text{Score}_{\text{r}} \) accounts for more than 50% of the superpixel area, we define the superpixel \( S_i \) as a reconstructed region. Otherwise, \( \text{Score}_{\text{d}} \) is used. If \( \text{Score}_{\text{d}} \) is very low compared with the reconstructed region, we classify \( S_i \) as a defective region.

4. Experiments

In our experiments, we first tested the proposed method in a simulated environment, before testing it in range of outdoor environments, with images taken from the ground and from a UAV. The UAV images were acquired using a DJI Phantom 2 equipped with a GoPro HERO 4 Silver (2560 \times 1920 px). The proposed method was demonstrated on a 3.3GHz Intel Core i5-4590 processor, 16GB RAM, and an Nvidia GeForce GTX 780 graphics card.

4.1 3D Reconstruction

We compared the Online SfM with VisualSFM \[26\], which is an offline SfM approach with excellent stability and accuracy. Figure 6 shows the qualitative result. Although Online SfM is noisier than VisualSFM, both of the reconstructed models are visually similar. Table 1 shows the quantitative result of the 3D reconstruction from UAV images. It took 204 seconds to acquire 120 images. Online SfM requires 169 seconds which is 5.45 times faster than VisualSFM because we fix the number \( N(= 5) \) of images for feature matching. Since VisualSFM performs feature matching for all images, whereas Online SfM performs only between \( N \) images, our approach is able to integrate a new image into the 3D map within a certain period of time except when a global bundle adjustment is performed.

4.2 Defect Detection Results

The goal of our method is to detect defective regions in the reconstructed model without user intervention. We use precision and recall for evaluating the line-based segmentation and defect
Table 1 Quantitative comparison between VisualSFM and OnlineSfM.

<table>
<thead>
<tr>
<th></th>
<th># registered images</th>
<th># 3D points</th>
<th>execution time (s)</th>
<th>average execution time per image (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisualSFM</td>
<td>120</td>
<td>13,776</td>
<td>1,258</td>
<td>10.48</td>
</tr>
<tr>
<td>Online SfM</td>
<td>88</td>
<td>16,800</td>
<td>169</td>
<td>1.92</td>
</tr>
</tbody>
</table>

Fig. 6 Qualitative comparison between VisualSFM and OnlineSfM.

Detection. In each process, these are calculated as follows:

\[
\text{precision} = \frac{TP}{TP + FP}, \quad (10) \\
\text{recall} = \frac{TP}{TP + FN}, \quad (11)
\]

where TP is true-positive, FP is false-positive, FN false-negative. The system was required to replicate a trial in which the user intervened. A higher recall indicated a more faithful replication.

To apply precision and recall, it is necessary to establish correct outcomes for each process. We prepared three datasets:

- **Simulation dataset** was a virtually simulated 3D structure.
- **Ground dataset** was taken from the ground.
- **Aerial dataset** was taken from a UAV.

Table 2 Simulation, ground and aerial dataset.

<table>
<thead>
<tr>
<th></th>
<th># images</th>
<th># labeled images</th>
<th>acquisition time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation</td>
<td>40</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Ground</td>
<td>53</td>
<td>10</td>
<td>106</td>
</tr>
<tr>
<td>Aerial</td>
<td>120</td>
<td>24</td>
<td>204</td>
</tr>
</tbody>
</table>

Table 3 Average precision and recall in each environment.

<table>
<thead>
<tr>
<th></th>
<th>Segmentation</th>
<th>Defect detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>precision</td>
<td>recall</td>
</tr>
<tr>
<td>Simulation</td>
<td>0.844</td>
<td>0.999</td>
</tr>
<tr>
<td>Ground</td>
<td>0.582</td>
<td>0.996</td>
</tr>
<tr>
<td>Aerial</td>
<td>0.508</td>
<td>0.989</td>
</tr>
</tbody>
</table>

Using Unity [27], we created a simulated environment containing only a single object, and without a complex background. As the target region was easy to acquire, we used this as the reference in the segmentation process. The reference for the defect detection process was based on the pixel-level difference between the reconstruction and the target region. In outdoor environments, reference outcomes were created at the superpixel level, because of the difficulty of identifying target regions and defects at the pixel level. These were created manually. Table 2 summarizes details of each dataset.

We measured the average execution time of detection of the defective regions per image. The segmentation process took 0.05 seconds, whereas defect detection took 0.04 seconds. Because the segmentation process is conducted in parallel with Online SfM, the essential execution time is only the defect detection time. The average acquisition time per image of Ground dataset is 2.00 seconds and that of the Aerial dataset is 1.70 seconds. As shown in Table 1, Online SfM took 1.92 seconds per image, so that execution time from an image input to defect detection is 1.96 seconds. Although there is a delay of 0.26 seconds in Aerial dataset, it is possible to present defective regions with almost no delay.

Table 3 summarizes the average precision and recall of our method in each environment. Here, segmentation refers to artificial object region segmentation, and defect detection to the overall performance. In the simulated environment, segmentation was almost perfect, but, some defective regions were not detected. In outdoor environments, each process demonstrated almost the same precision and recall as in the simulated environment. The results suggest that the proposed method worked as well in the outdoor environments as it did in the simulated environment.

We compared the proposed method with online feedback [12]. Our method was divided into two parts, that is, line-based segmentation and defect detection consisting of Scorea and Scorera. We also compared the following three methods: our proposed method, our method without segmentation and our method without Scorera. Because the online feedback approach requires 3D surfaces to evaluate the quality of the 3D model, we generate 3D surface from 3D points by the ball pivoting algorithm [28] using MeshLab [29]. In the online feedback approach, each mesh has a quality score, and a user checks...
for defective regions in a 3D model by visually inspects. We project each mesh to a 2D image and judge defective regions by varying the threshold value of the quality score. Additionally, we vary the threshold value of our method.

Figure 7 shows the receiver operating characteristic (ROC) curve of defect detection in outdoor environments. The proposed method with segmentation outperforms the online feedback approach. Because online feedback evaluates the quality of 3D models based on only reconstructed 3D points, defect detection is likely to fail when only few points exist around the target region. The result also shows both online feedback and our method without segmentation are similar in terms of performance. In contrast, by identifying target regions by line-based segmentation, it is possible to detect defects in regions where there are few points but essential target regions. Compared to online feedback, our approach is more useful and helpful for the user because it detects defective regions in the 3D reconstruction without user intervention.

Figure 8 shows examples from each dataset: (a) Simulation dataset, (b) Ground dataset, and (c) Aerial dataset. In Fig. 8 (a) (b) (c), each row shows the last image in sequence, reprojected 3D points to the last image, and defective regions (filled superpixels), respectively. As the row proceeds from the top to the bottom, the number of images in sequence increases. Thus, the defective regions gradually disappear and our method successfully follows the changes. Note that a naive segmentation often extracts unwanted regions, such as the boundary between buildings and the sky, whereas our method is able to detect defective regions only on the artificial structures. Figure 9 shows the 3D points viewed from a viewpoint in front of the building. Note that there are several regions with sparse points, that suggests further images should be taken to complete the dense 3D reconstruction. Our method is helpful to inform the operator of such regions.

5. Conclusions

We proposed a method that facilitates for the identification of defective regions in 3D reconstruction during the process of image acquisition, using online SfM to reconstruct the 3D model. Because artificial structures commonly comprise line segments, we used these to extract the target region. Any defective region is then identified from the 3D structure and presented on a newly acquired image to help the UAV operator comprehend the reconstruction state. Our method was experimentally demonstrated to work in a number of outdoor environments. In future work, we will extend this to a wider range of
environments. We will also extend our method of automatic image acquisition using UAVs to further improve 3D reconstruction by developing processes for tracing the cause of defects and estimating a retake position.

References


Seiya Ito
He received his B.S. degree from Aoyama Gakuin University, Japan, in 2016. He is currently a M.S. student at Aoyama Gakuin University. His research interests include computer vision.

Naoshi Kaneko
He received his B.S. and M.S. degrees from Aoyama Gakuin University, Japan, in 2012 and 2014, respectively. He is currently working toward the Ph.D. degree in the School of Science and Engineering, Aoyama Gakuin University. His research interests include computer vision and human sensing.

Takeshi Yoshida (Member)
He received his B.S., M.S., and Ph.D. degrees from Kobe University, Japan, in 2007, 2009, and 2013, respectively. In 2015, he joined Aoyama Gakuin University, where he is currently an Assistant Professor of the Department of Integrated Information Technology. His research interests include computer vision and robot vision.

Kazuhiko Sumi (Member)
He received his B.S., M.S., and Ph.D. degrees from Kyoto University, Japan, in 1982, 1984, and 1997, respectively. In 2011, he joined Aoyama Gakuin University, where he is currently a Professor of the Department of Integrated Information Technology. His research interests include computer vision for automation, surveillance, and biometrics.