Generation of Building Facade from Sequential Urban Images for Polygonal Buildings on 3D Map

HIDEAKI ORII* Member, HIDEAKI KAWANO† Non-member
HIROSHI MAEDA† Non-member

(Received February 18, 2013, revised April 15, 2013)

Abstract: In this paper, a method to generate building wall textures from an in-vehicle camera is proposed as an aid to construct 3D maps. The building wall textures are required to attach to 3D polygons which are obtained through 3D measurements in urban space. We assume that the in-vehicle camera is under linear uniform motion and building walls are planar regions perpendicular to the optical axis. Under the assumption, the same building wall region has the same depth, or disparities, over the region among successive images. Since disparities derived from foreground objects are different from the disparities derived from the building wall, we can use the disparity differences as a clue to effectively distinguish the building walls from the foreground objects. We formulate the extraction of building wall textures incorporating these considerations as an optimization problem which can be solved by graph-cuts algorithm. To show the effectiveness of the proposed method, it is applied to a sequential scene in a miniature model street and an actual street.

Keywords: building facade, graph cuts, 3D map

1. Introduction

Three dimensional maps are widely used in recent years, and have become more important in car navigation systems and simulation systems for a city planning. On the other hand, because works for making the 3D maps require a lot of time and the labor, it is necessary for sustainable updatings and covering broader regions to automate or assist a construction of the 3D maps. As a part of such efforts, street views are often collected by using in-vehicle cameras. The collected sequential images are utilized to model 3D building walls. In many cases, indeed those collected images contain building wall textures which are required to model the city space, but undesired foreground objects such as street trees, pedestrians, parking cars, power poles, etc. are also involved. Those foreground objects could have harmful effects in modeling or obtaining textures for 3D maps.

Some studies that extract the region of the specific object in the image have been proposed. In the studies [1]~[4], extraction of objects is achieved using projection matrix and posture of the objects which are calculated by using some corresponding feature points between images. In the studies [5]~[9], planar regions are estimated from similarity between projective transformation images. Recently, Taeda et al.[10] proposed the planar objects extraction method using Graph-Cuts segmentation.

In this paper, a method to extract building wall textures from an in-vehicle camera is proposed as an aid to construct the 3D maps. The building wall textures are required to attach to 3D polygons which are obtained through 3D measurements. To achieve the extraction of building wall textures from the images taken by in-vehicle camera, our approach is based on utilizing the planar assumption for extracted object. Assume that the in-vehicle camera is under linear uniform motion; the same building wall with the same depth has the same disparities between successive images. Since disparities derived from foreground objects are different from the disparities derived from the building wall, we can use the disparity differences as a clue to effectively distinguish the building walls from the foreground objects. We formulate the texture extraction method incorporating these observations as an optimization problem which can be solved by graph-cut algorithm. To show the effectiveness of the proposed method, it is applied to scenes taken on the rail in a model street.

The proposed method focuses on the following key observations, which are also schematically explained in Fig.1:

- Foreground objects have comparatively smaller depths than building wall surfaces, and as a result, disparities for the foreground objects derived from successive images are bigger than ones for the building walls.
- The disparities of building walls between successive images are constant when the camera is under uniform motion and looks towards buildings exactly.
• In each image, regions of building wall are dominant against for regions of foreground objects.

When successive two images are superimposed by the dominant disparity which comes from building walls, we can make use of misalignments which arises only at regions other than building walls in order to segment building wall textures from the images.

2. Proposed Method

In this section, we describe the detail of the proposed method. To generate good textures of a polygonal building on 3D map, the textures are composed from temporal sequence urban images taken by an in-vehicle camera in proposed method. When the image sequence is taken, the in-vehicle camera is installed sideways in the direction of forward movement. At this time, the positional coordinates of a car with a global positioning system (GPS). Figure 2 shows the overview of proposed algorithm.

The proposed method is divided into two main process, i.e. Estimate plane region, Generate texture of plane. In first process, building wall regions in urban images are extracted as plane regions to omit unwanted objects for generation of texture, such as a pedestrian. The Extraction of building wall plane region is formulated as a energy minimization problem based on the difference between sequences of urban image, and the minimization problem is solved with graph cut. In second process, textures of a polygonal building are generated with the patches of extracted region. The correspondence of a polygonal building and a patch is determined by information GPS and 2D map. In the following, the detail of each process is concretely explained.

2.1 Estimation of Plane Region

2.1.1 Estimation of Disparity

As an initial setting phase, the disparity for building wall is estimated. At first, a pair of successive images are selected to find corresponding points using SIFT features[11]. For each corresponding points, a disparity can be calculated. However, there are corresponding points that are from foreground object such as a pedestrian or a street tree. In this paper, the disparity of a building wall is estimated with simple method that is derived from a geometrically restriction. Assuming the area of the foreground objects occupies less than that of the building walls, the median among the all disparities calculated for all corresponding points can be considered as an estimation of disparity for building wall as follows.
\[ d = \text{median}_{i=1,2,\ldots,n} | P_{x_i} - P_{x_i'} | \]  

Here, let \( d \) be a disparity, and let \( n \) be a number of corresponding points. \( P_{x_i} \) and \( P_{x_i'} \) (\( i = 1,2,\ldots,n \)) are horizontal coordinate of corresponding points in successive images. When the camera is under uniform motion and looks towards buildings exactly, we can apply the estimated disparity to all images. An arbitrary pair of two successive images \( I_1 \) and \( I_2 \) is superimposed by using the estimated disparity. As a result, two duplicative regions \( u_1 \) and \( u_2 \) from both two images can be obtained.

### 2.1.2 Extraction of Building Wall Region

On the basis of image information with respect to \( u_2 \), a label which represents “building” or “others” is assigned to each pixel in \( u_1 \). Concretely, we make use of the fact that there exists the same region on the site of pixel \((x+d,y)\) in \( u_2 \) if an arbitrary pixel \((x,y)\) in \( u_1 \) belongs to “building”. Symbol \( d \) denotes the disparity.

If a label is “building” Assume \( S_u \), a label assined to a pixel \( u \), label \( S_u \) is defined as follows:

\[ S_u = \begin{cases} 
1 & \text{(building)} \\
0 & \text{(other)} 
\end{cases} \]  

(2)

We optimize the function as follows:

\[ E(S_u) = \sum_{w \in \text{ROI}} E_{\text{data}}(S_u) + \sum_{(u,v) \in N} E_{\text{smooth}} \]  

(3)

where \( E_{\text{data}} \) is a cost term representing plausibility as buildings, \( E_{\text{smooth}} \) is another cost term with respect to adjacent pixels \( u,v \). Here, \( E_{\text{data}} \) is defined as follows:

\[ E_{\text{data}} = \begin{cases} 
\sum_{i,j} \left| u_1(i,j) - u_2(i,j) \right|^2 & \text{if } S_u = 1 \\
\sum_{i,j} \left| u_2(i,j) \right|^2 & \text{if } S_u = 0 
\end{cases} \]  

(4)

A variable \( E_{\text{smooth}} \) works as a threshold to decide the label of pixel \((i,j)\). If difference between respective local image patch in image \( u_1 \) and \( u_2 \) is lower than the threshold, the label of pixel \((i,j)\) tends to be assigned to “building”.

And \( E_{\text{smooth}} \) is defined as follows:

\[ E_{\text{smooth}} = \begin{cases} 
0 & \text{if } S_u = S_v \\
\exp(-|u-v|) & \text{if } S_u \neq S_v 
\end{cases} \]  

(5)

\( E_{\text{smooth}} \) leads adjacent pixels to be assigned to the same label if the intensities of adjacent pixels are similar. The cost function composed of above two cost terms are minimized by graph-cut algorithm[12][13]. In graph-cut algorithm, this energy minimization problem is reduced to instances of the maximum flow problem in a graph as shown in Fig.3. Because adjacent pixels with similar intensities tend to be the same label by the work of \( E_{\text{smooth}} \), we can obtain cohesive segmentation results as shown in Fig.4.

### 2.2 Generation of Textures

In this section, the process for generation of the texture corresponding to a polygonal building on 3D map from the patch images of extracted building wall region is described in detail. In proposed method, a building wall region in an image is separated into patches of a given size, after images that contain target building are found from temporal sequence urban images. Then the texture of target building wall is synthesized with those patches.

#### 2.2.1 Extraction of Patches

First, frontal shots of target building are found based on the coordinate information of GPS and the information of 2D map. The width of a building in an image could be estimated by the actual width and the depth of that building as shown in Fig.5. The width \( [\text{pix}] \) of a building can be calculated as follows.

\[ x_{\text{width}} = \frac{X \times f \times x}{Z \times C} \]  

(6)
Here, let $x_{\text{width}}$ be the width [pix] of a building in an image, and let $X$ and $Z$ be the actual width [m] and the depth of a building. Let $f$ and $C$ be the focal length and the size of imaging element of a camera. Thus, extracted building wall region could be aligned a correspondent polygonal building on 3D map. However, it is difficult to identify building region with accuracy, because the information of GPS and the information of 2D map have a margin of error. Therefore, following process is performed. Next, the patches to synthesize polygonal building wall texture are extracted from building regions in correspondent image. The patches are extracted from the center region of frontal shots of target building at first, and other patches are extracted based on the similarity between those and the center patches. The similarity (the bhattacharyya distance) is calculated as follows.

$$ R(P(x_1), P(x_2)) = \sqrt{1 - \frac{\sum \sqrt{h(x_1)h(x_2)}}{\sum h(x_1) \cdot \sum h(x_2)}} $$

(7)

Here, let $P x_i$ be a patch center on $x_i$, and let $h(x_i)$ be a histogram of the Hue value of pixel in $P x_i$.

2.2.2 Fusion of Patches The texture of a target building wall is generated by the texture mapping with extracted patch images. The extracted patches contain the overlapping region and the missing region, because the patches are extracted from more than one image. In this paper, the textures are synthesized by graph cuts energy minimization with multi labels for the fusion of these patches.

The objective function for the texture synthesis is defined as follows.

$$ I_{\text{eng}} = \sum I_{\text{data}}(l_p) + \sum I_{\text{smooth}}(l_p, l_q) $$

(8)

Here, let $l_p$ and $l_q$ be a pixel value of the texture. $I_{\text{data}}()$ is a cost term representing plausibility as building textures, $I_{\text{smooth}}()$ is an another cost term with respect to adjacent pixels $l_p, l_q$. $I_{\text{data}}()$ is defined as follows.

$$ I_{\text{data}}(l_p) = 1 - \text{Hist}(l_p) \frac{\sum_{l_p} I_{\text{bin}}^p}{\sum_{l_p} I_{\text{max}}} $$

(9)

Here, Hist() is the frequency of a bin of a three dimensional histogram with respect to a pixel $p$. let $I_{\text{bin}}^p, I_{\text{max}}^p$,
the histogram. Let $I_{\max}$ be the max value of a pixel. Let $I_{\text{bin}}$ be the number of bins of the histogram. $\text{Hist}(\cdot)$ range from 0 to 1, because the sum of the value of bins of histogram is normalized. Thus, energy becomes small when a color value with higher frequency. $I_{\text{smooth}}()$ is defined as follows.

$$I_{\text{smooth}}(I_p, I_q) = \frac{I_R(p, q) + I_G(p, q) + I_B(p, q)}{3}$$ (10)

$I_{\text{smooth}}()$ act to let labels are stationary on pixels that have small difference. $I_R(p, q)$, $I_G(p, q)$, $I_B(p, q)$ are defined as follows.

$$I_R(p, q) = |I_R^p(p) - I_R^q(p)| + |I_R^p(q) - I_R^q(q)|$$ (11)

$$I_G(p, q) = |I_G^p(p) - I_G^q(p)| + |I_G^p(q) - I_G^q(q)|$$ (12)

$$I_B(p, q) = |I_B^p(p) - I_B^q(p)| + |I_B^p(q) - I_B^q(q)|$$ (13)

Here, $p$ and $q$ are coordinates of adjacency pixels. $I_p$ and $l_q$ are labels with respect to a pixel $p$ and a pixel $q$. $I_R(p, q)$, $I_G(p, q)$, $I_B(p, q)$ are a sum of a value of pixel $p$ and a value of pixel $q$. This objective function is solved by graph cuts energy minimization with $\alpha$-expansion [12][13].

2.2.3 Completion of Textures In proposed method, the lack of the textures is suppressed to an extent because the patches are obtained from temporal sequence images. However, it is difficult to avoid the lack of the textures. Therefore, a lack texture is filled with a presumable texture of other area like image completion method proposed by Wexler et. al.[14]. In this paper, restricted search of a presumable texture based on a knowledge of a building texture is performed. In particular, searching is performed in a vertical or a horizontal direction with respect to a target patch. Figure 7 shows the overview of restricted search. Here, let $\Omega$ be a lack region, and let $\Omega'$ be an extended region of $\Omega$. In Figure 7, the search target region with respect to pixel $x_p$ is yellow painting area.

3. Experimental Results

To verify the effectiveness and validity of the proposed method, the proposed method is applied to a miniature street and an actual street image in the experiment.

3.1 Experiment for Miniature Street In this experiment, a miniature street model with about 1/100 scale of the actual size is used. For example, A building of 30 meters with 10 stories is represented by a 30-centimeter-high box model.

Figure 8 shows the experimental setting of the miniature street model and Table 1 shows the spec of the camera. In the experiment, the obstacles were set up in front of the miniature of buildings, and we took pictures at a fixed time interval with keeping the camera be perpendicular to the direction of travel and be moved at the constant speed. Figure

<table>
<thead>
<tr>
<th>camera</th>
<th>NikonD100</th>
</tr>
</thead>
<tbody>
<tr>
<td>size of imaging element</td>
<td>23.7x15.6mm</td>
</tr>
<tr>
<td>focusing length</td>
<td>24mm</td>
</tr>
<tr>
<td>resolution of image</td>
<td>1280 × 960</td>
</tr>
</tbody>
</table>

The table above shows the spec of the camera used in the experiment. The size of the imaging element is 23.7x15.6mm, the focusing length is 24mm, and the resolution of the image is 1280 × 960.

Figure 9: Some sampled images in the sequential images at the experimental settings. Figure 10: Experimental results for the conventional method [10]. Blue painting areas are the estimated building wall textures. Figure 11: Experimental results for the proposed method. Blue painting areas are the estimated building wall textures.
9 are some sampled images in the sequential images at that condition.

First, the building wall regions are estimated. Figure 10 shows the experimental results for applying the conventional method [10] to previously mentioned scenes. In Fig. 10, blue painting means the building wall textures estimated by the conventional method. Figure 11 shows the experimental results for applying our proposed method to previously mentioned scenes. In the images, blue painting means the building wall textures estimated by our proposed method. As we can see in Fig. 11, simulated city street images were segmented into building wall textures and the others effectively. As we can also see in Fig. 11, indeed the region estimation doesn’t work well around the complicated boundaries, but it doesn’t become a matter when we apply texture synthesis method for the purpose of foreground elimination.

And Table 2 shows the estimation result of the width of building in an image. The proposed method can estimate the width [pix] with high accuracy as evidenced by the table.

Finally, the textures of building walls are generated by the estimated building wall region texture. Figure 12 shows the estimated building wall region and Fig. 13 shows the result of texture synthesis.

### 3.2 Experiment for Actual Street

In this experiment, the proposed method is applied to images of an actual street. Figure 14 shows the 2D map of the actual street and Table 3 shows the spec of the camera that is used, and Table 4 shows the spec of GPS that is used.

Figure 15 are some sampled images in the sequential images at that condition. And Fig. 16, Fig. 17, Fig. 18, Fig. 19 are the experimental result of each process of proposed method. As shown in Fig. 19, the textures of a building are generated by the proposed method.

### 4. Conclusion

In this paper, a method to generate building wall textures from an in-vehicle camera was proposed. Our approach is
based on utilizing the planar assumption for extracted object, and Assume that the in-vehicle camera is under linear uniform motion; the same building wall with the same depth has the same disparities between successive images. We formulated the texture extraction method incorporating these observations as an optimization problem which can be solved by graph-cut algorithm. The effectiveness of the proposed method was verified applying the method to scenes taken in a model street and an actual street.

References


Acknowledgment
A part of this work was supported by Grant-in-Aid for JSPS Fellows KAKENHI Grant Number 12J09839.