ABSTRACT
Extracting interest points is one of the most important issues in techniques such as an object recognition and classification or a place recognition. Correct feature extraction can efficiently find out which parts of the image are likely to be unique, and can also match similar or identical parts in two different images. However, previous feature extraction algorithms typically extract features based on corner points with a large gradient, which results in degraded performance in less textured images, such as flat walls. In this paper, we propose a novel feature extraction method that can extract points in the texture-less images. Our method extracts interest points from the contour region of neighboring superpixels, which has a little or more changes of color values, then use a BRIEF descriptor to extract low-dimensional descriptors. Experimental results show that our method can extract features from texture-less regions and perform the matching between two different images with a high accuracy.

KEYWORDS
Interest Point Detection, Feature Extraction, Superpixel Segmentation, BRIEF Descriptor, Place Recognition

1 INTRODUCTION
Recently, research on methods of recognizing or classifying objects or places in images has been actively studied. For example, objects in these techniques can be used for real-time tracking, recognize three-dimensional objects in augmented reality, and recognize pedestrians or obstacles in autonomous driving. Furthermore, these techniques also can be applied to a method of semantic segmentation to classify and recognize a specific object or person existing in a two or three-dimensional image semantically. In the simultaneous localization and mapping (SLAM), place recognition is used for loop closure detection, that has a function to recognize when an input scene image is the same place as a previously revisited ones. One of the most important techniques in implementing such recognition and classification methods is feature extraction. The more accurately the correct features are extracted early, the higher the accuracy of recognition and classification.

SIFT [8], SURF [2], and ORB [10] are typical examples of such feature extraction methods. These methods mainly extract the features of the image by detecting a corner point and obtaining a descriptor for the corner point. However, the descriptors that can be obtained through these methods are usually stored in the form of a high-dimensional vector of about 128 dimensions or a storage space of 64 bytes or more. In addition, since the feature extraction algorithms described above extract feature points based on a region having a large amount of change, there is a disadvantage in that it is not possible to extract feature points abundantly in an image having a small variation such as a flat wall.

In this paper, we use the newly proposed feature descriptor extraction algorithm instead of SIFT, SURF, ORB. In the proposed method, images are clustered to the similar size of superpixels through the SLIC method [1]. Next, the feature points are detected at the boundaries of the superpixels, because the boundaries have a small or large gradient value. Then, using the BRIEF [3] algorithm through the detected feature points, a binary descriptor is generated.

2 INTEREST POINT DETECTION
2.1 Simple Linear Iterative Clustering (SLIC)
Superpixel can be defined as a group of pixels which have similar characteristics. The position similarity and a color similarity are the most commonly used characteristics. Superpixel can be very helpful for image segmentation because of these characteristics for the segmentation process. Figure
2 shows results of superpixel segmentation using different numbers of segmentation. But in our method, we don’t set the specific number of segmentation, so the number of segmentation depends on the image.

Simple linear iterative clustering (SLIC) [1] is state of the art superpixel segmentation algorithm with a low computational overhead. To cluster similar pixels and segment superpixels, SLIC method requires 5-dimensional features, 3 components are 3-channel color values and 2 components are 2-dimensional pixel position. Generally, this method just requires the number of superpixels as a parameter and creates a number of superpixels similar to this parameter.

In SLIC algorithm, they use a CIE-Lab color space because CIE-Lab color space is not affected by the display device or printer, unlike an RGB color space. So we have to convert the RGB color space to the Lab color space [5]. In this process, RGB color space should have converted to the CIE-XYZ color space first because we can’t convert an RGB color space to the CIE-Lab directly. Eq. 1 shows equations of transformation:

\[
\begin{align*}
X &= 0.4303R + 0.3394G + 0.1505B \\
Y &= 0.2126R + 0.7152G + 0.0722B \\
Z &= 0.0193R + 0.1192G + 0.9505B
\end{align*}
\]  

After that, we can convert to CIE-Lab color space using following equations:

\[
\begin{align*}
L^* &= 116f\left(\frac{Y}{Y_0}\right) - 16 \\
a^* &= 500f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \\
b^* &= 200f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right)
\end{align*}
\]

where

\[
f(q) = \begin{cases} 
\sqrt{q} & (q > 0.008856) \\
7.787q + \frac{16}{116} & (q \leq 0.008856)
\end{cases}
\]

And we can’t use a general Euclidean distance as a distance measurement method because we should normalize not only the color features but also the spatial features. So Achanta et al. [1] propose a different way of distance measure to cover the size of superpixel. SLIC algorithm makes a similar size of superpixels each other, so it is important to cover spatial features. When we know the number of pixels and superpixels, we can estimate the approximated size and a grid interval of average superpixel using following equations:

\[
A = \frac{N}{K}
\]

\[
S = \sqrt{A}
\]

where \(A\) is the spatial extent of superpixel, \(S\) is a grid interval, and \(N \) and \(K\) are the number of pixels and superpixels, respectively. After that, they set \(m\) as a perceptual color distance threshold value. This \(m\) utilizes for the weighting of distance equation. If a spatial pixel distance is bigger than \(m\), we can gain more weights to the color similarity based on Eq. 8. And now, we can calculate the distance from the center \(C_k = [l_k, a_k, b_k, x_k, y_k]\) to targets using following equations:

\[
\begin{align*}
\mathbf{d}_{\text{lab}} &= \sqrt{(l_k - l)^2 + (a_k - a)^2 + (b_k - b)^2} \\
\mathbf{d}_{xy} &= \sqrt{(x_k - x)^2 + (y_k - y)^2} \\
D &= \mathbf{d}_{\text{lab}} + (\frac{m}{S})\mathbf{d}_{xy}
\end{align*}
\]

where \(k = [1, K]\) and \(d_{\text{lab}}\) and \(d_{xy}\) are the Euclidean distance of color features and spatial features, respectively. As a result, the greater the value of \(m\), the more spatial proximity gets weight and the more compact the cluster. In [1], they have chosen \(m=10\).

Based on the measured distance, SLIC makes a similar size of superpixels each other. First, \(K\) regularly spaced cluster centers are set. Second, find the lowest gradient position into the 3x3 neighbors from the center. The gradient \(G(x, y)\) is calculated by the \(L_2\) Norm-based Eq. 9:

\[
G(x, y) = ||I(x + 1, y) - I(x - 1, y)||^2 + ||I(x, y + 1) - I(x, y - 1)||^2,
\]

where \(I(x, y)\) is the Lab vector corresponding to the pixel position \((x, y)\). Each pixel of an image is related to the nearest cluster center which is overlapped with the search area. This search area is defined by \(2S \times 2S\). After all the pixels are associated with the nearest cluster center, a new center is computed as the average Labxy vector of all the pixels belonging to the cluster. An entire process is performed iteratively until the residual error converges. The residual error is the \(L_1\) distance from the previous center to new ones.

### 2.2 Intermediate Point Detection

Given results of superpixel segmentation, we made a local patch based on the intermediate point of two superpixel centers. To get the center of superpixel, we made use of the concepts of image moments [4]. In physics, if the points represent mass, then the zeroth moment is the total mass, the first moment divided by the total mass is the center of mass, and the second moment is the rotational inertia. The mathematical concept is closely related to this concept. In mathematics, the \(n\)-th moment about the point \(c\) is defined as:

\[
\mu_n = \int_{-\infty}^{\infty} (x - c)^n f(x) dx
\]

But we have to consider the 2-dimension because we will cover the image. So we can modify Eq. 10 to the 2-dimensional form about the point \((c_x, c_y)\):

\[
\mu_{m,n} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - c_x)^m (y - c_y)^n f(x, y) dy dx,
\]

and this equation should change to the discrete form because the image consists of discrete pixels. So, finally we can get an \(m, n\)-th moments of a 2-dimensional image about the point \((c_x, c_y)\):
Interest Point Detection using the Superpixel Segmentation and a Binary Descriptor

\[ \mu_{m,n} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} (x-c_x)^m (y-c_y)^n f(x,y), \]  

(12)

where \( f(x,y) \) is the actual image. Using this mathematic concept, we can calculate the area of a binary image using 0th moment. In this case, \( x^0 \) and \( y^0 \) don’t have any effect. So we can remove them and define the area of a binary image like Eq.13:

\[ \mu_{0,0} = \sum_{x=0}^{w} \sum_{y=0}^{h} f(x,y) \]  

(13)

where \( w \) and \( h \) are the width and height of an image. And we can also calculate the centroid of a binary image using 0th and 1st moment:

\[ \text{centroid} = \left( \frac{u_{1,0}}{u_{0,0}}, \frac{u_{0,1}}{u_{0,0}} \right) \]  

(14)

Using the concept of moments, we can calculate the area and centroid of a binary image. In this paper, we changed the result of superpixel segmentation to a binary image. Each contour is set to 1 and other regions are set to 0. Using this binary image, we can get the centroid of each superpixel.

In this paper, we get a local patch and its descriptor based on the intermediate point of two neighboring superpixel centers calculated from the image moments. First, the nearest neighbor superpixel is required. We found the nearest neighbor superpixel based on the horizontal axis only. Of course, the vertical axis can also be considered. But if we use both horizontal and vertical axis, too many information is acquired and it causes the increases in computational costs. After we found what the nearest neighbor superpixel is, two centroids of the superpixel and its neighbor are connected and find the intermediate point between two centers based on the connected straight line. Figure 3 briefly shows the intermediate point extraction.

Superpixel consists of pixels that have a very similar color and similar position because the segmentation algorithm is based on the gradient ascent. That is, in the contour of superpixels, we guarantee there is a big or small change of color value. So we made use of this change from the contour. Generally, previous feature extraction algorithm is based on the big change. But using the proposed method, we can extract features both big and small changes of color value. As a result, we can make use of the proposed method to the place recognition especially indoor scenes because we can extract features from the textureless walls or ceilings.

2.3 BRIEF Descriptor

The final process of local patch descriptor generation is to make a descriptor from local patches that we made from the previous process. In order to compensate the degraded computational speed because of the superpixel-based feature detection, we use the BRIEF descriptor, which is faster than the SIFT and SURF in the matching process [3]. The binary descriptor vector made by the BRIEF method only requires
where \( n \) is generally set to 256 or 128.

This method starts from the assumption that we can classify the patch just comparing a few pairs of a pixel, among all the possible pairs of a pixel into the patch. The following equation is the test function to compare two pixels in the patch \( p \):

\[
\tau(p; x, y) = \begin{cases} 
1 & p(x) < p(y) \\
0 & \text{otherwise}
\end{cases}
\] (15)

Here, \( p(x) \) means the pixel value at the position \( x = (u, v)^T \) and this \( x \) is the position in the patch. This patch is smoothed by the kernel. And we select \( n_d \) number of pairs in the patch without any duplicate and calculate the \( n_d \) bit descriptor using Eq. 16:

\[
f_d(p) = \sum_{1 \leq i \leq n_d} 2^{i-1} \tau(p; x, y)
\] (16)

4 CONCLUSION

In this paper, we propose a feature extraction method that can detect rich feature points in a texture-less region and produce them in the form of small capacity descriptors. To detect the interest points, we first performed superpixel segmentation through SLIC, and based on the created superpixels we detected two centers of neighboring superpixels and regards the midpoint between them as a feature point. After that, we could obtain binary-based 256-bit descriptors using the BRIEF descriptor. In experimental results, our method obtained an average of 6 matching pairs compared to SIFT and 10 matching pairs on average compared to SURF.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP) (No. 2011-0030079).
Table 1: Comparison of the number of matching lines

<table>
<thead>
<tr>
<th></th>
<th>SIFT</th>
<th>SURF</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>livingroom0</td>
<td>51 pairs</td>
<td>48 pairs</td>
<td>55 pairs</td>
</tr>
<tr>
<td>livingroom1</td>
<td>53 pairs</td>
<td>47 pairs</td>
<td>61 pairs</td>
</tr>
<tr>
<td>copyroom</td>
<td>44 pairs</td>
<td>51 pairs</td>
<td>48 pairs</td>
</tr>
<tr>
<td>office1</td>
<td>28 pairs</td>
<td>23 pairs</td>
<td>32 pairs</td>
</tr>
</tbody>
</table>

REFERENCES


