Fast and Robust Building Extraction Based on HSV Color Analysis Using Color Segmentation and GrabCut

Takuya Futagami *, Noboru Hayasaka **, and Takao Onoye *

Abstract: In this paper, we propose a method for automatically extracting buildings from scenery images. The method utilizes color segmentation and GrabCut on the basis of the fact that the background regions of scenery images tend to be found in the upper and lower parts of the images. We evaluated its extraction accuracy and computational time by using 106 high-quality scenery images (HQ dataset) and 89 low-quality ones (LQ dataset). Experiments showed that $k$-means clustering for color segmentation and HSV color space allow the proposed method to achieve higher extraction accuracy and faster computational time compared with the conventional method. The proposed method improved the extraction accuracy by 14% or more and reduced the computational time by 5% or more for both datasets compared with the conventional method. Comparing the extraction accuracy of the proposed method by using different color spaces, HSV color space improved the accuracy by more than 2.78% for the LQ dataset due to its noise robustness. The experiments, however, suggest that the proposed method has room for improvement in terms of the process of generating the initial seed used to initialize GrabCut.

Key Words: scenery image, building extraction, GrabCut, color segmentation, building recognition.

1. Introduction

In recent years, navigation applications, which guide users to their destination, have been widely used as the number of smartphone users increases rapidly. Navigation systems track a user’s position by means of global navigation satellite system (GNSS) signals [1]. The accuracy of GNSS in urban areas, however, decreases because of signal reflection caused by surrounding buildings [2].

Systems that track a user’s position by means of recognizing buildings from a scenery image captured by mobile devices are being proposed as an alternative to GNSS [3],[4]. As for studies on building recognition, Åhlen et al. removed parts of the background not containing buildings as pre-processing to improve the recognition accuracy [5]. They reported that the recognition performance depends on the accuracy of building extraction. The accuracy of automatic building extraction, therefore, must be increased in order to improve the building recognition performance. Additionally, automatic building extraction is expected to be developed for inspecting building surfaces or reconstructing 3D buildings [5],[6].

Ueno et al. developed a method for automatically extracting buildings that uses GrowCut [7] initialized with seeds generated on the basis of binarization [6]. Ueno et al.’s method improved the extraction accuracy by 40% or more compared with Åhlen et al.’s method [5]. Hereafter, we refer to Ueno et al.’s method [6] as the conventional method in this paper. The conventional method, however, tends to have a high rate of wrongly determining background regions as building regions. This tendency is caused by the fact that they tried to avoid wrongly determining building regions to be background regions. Hence, we propose a method for automatically extracting buildings and confirm the effectiveness of it.

Deep-learning-based semantic segmentation networks [8] such as SegNet [9] have the possibility of extracting building regions accurately. For example, several studies exploit SegNet for a semantic segmentation which can separate building regions from background regions [9],[10]. The training cost of the semantic segmentation network, however, is known to be much higher than other machine learning algorithms [11]. Although transfer learning [12] and data augmentation [13] are expected to reduce the cost of annotation, it is difficult to prepare a sufficient amount of annotated training data for a wide variety of building or background types and shapes. We, therefore, develop automatic building extraction that requires no annotated training data.

When a scenery image is captured by looking toward and up from the ground, it tends to have background regions (e.g., sky, pedestrians, and road) at the top or bottom. The proposed method is designed to extract building regions under the assumption that scenery images, whose main subjects are buildings, have this tendency. This assumption allows the proposed method to forgo annotated training data. It uses color segmentation, which generates color clusters on the basis of a clustering technique, and GrabCut [14], which is an interactive foreground extraction technique.

In references [6],[15], the quality of scenery images used for evaluation, which are included in the Zurich building database (ZuBuD) [16], is not low since they are captured during sunlight hours. The extraction accuracy for low quality images, however, has not been evaluated. Moreover, the processing speed of each building extraction method in these references has not been evaluated. Towards quickly, accurately, and effectively extracting buildings in various environments, we will...
evaluate the extraction accuracy for low quality images and the computational time.

Our previous method [15] extracts building regions by using only RGB color space. References [17]–[19], however, suggest that RGB color space is not always suitable for image segmentation tasks. We, therefore, experimentally determine a suitable color space for extracting buildings to improve the extraction accuracy compared with our previous method [15].

The rest of this paper is described as follows. In Section 2, related work is described, and in Section 3, the proposed method is outlined. In Section 4, the experiments done to show the effectiveness of the proposed method are presented, and in Section 5, a discussion is given. The present study is summarized and concluded in Section 6.

2. Related Work

2.1 Clustering Method

The proposed method uses color segmentation based on color clustering in order to divide building and background regions into different segments. Which clustering method is suitable in terms of both extraction accuracy and computational time, however, was not verified in the preceding study [15]. Hence, a suitable method is experimentally determined in this paper. In this section, we summarize the four basic clustering methods that we set as evaluation targets.

2.1.1 K-means

Practically, k-means [20] is known as the most popular clustering method and is categorized as a partitioning-based algorithm [21]. It updates the centers of clusters (centroids) to minimize the distance to each data point from the centroids. The number of clusters \( k _ { m } \) is set as a parameter.

Since k-means is a simple and fast algorithm, it is suitable for large datasets. However, the effectiveness of k-means for image segmentation in building extraction from scenery images has not yet been confirmed in the literature. Hence, we experimentally determine which clustering methods are more suitable in terms of both extraction accuracy and computational time. We introduce other basic clustering methods in Sections 2.1.2 and 2.1.3.

2.1.2 Mean-shift

Mean-shift [22] is categorized as a density-based algorithm [21] and uses a kernel density estimator repeatedly. Although mean-shift automatically determines the number of clusters, bandwidth \( h \) is set as a parameter. The quality of clustering, however, decreases when the density of the data space is not uniform [23]. Additionally, the computational time of mean-shift is higher than that of k-means [24].

2.1.3 GMM/VBGMM

The Gaussian mixture model (GMM) [25] and variational Bayesian Gaussian mixture model (VBGMM) [26] are categorized as model-based algorithms [21] and use a GMM estimated by the expectation maximization (EM) algorithm. The number of mixture components \( k _ { v } \) is set as a parameter of VBGMM. VBGMM finds the number of clusters automatically.

GMM and VBGMM assume that the density of each cluster is modeled as a Gaussian distribution. This assumption, however, is not always true [27]. Additionally, the computational time of GMM and VBGMM is higher than that of k-means [24].

2.2 Color Space

The conventional method [6] and our previous method [15] use only RGB color space for extracting building regions. Which color space is suitable for extracting buildings, however, has not been verified yet. Hence, we experimentally determine which space is suitable as is the case with references [17]–[19] in order to improve the extraction accuracy. In this section, we summarize the five basic color spaces that we set as evaluation targets.

2.2.1 RGB

Red, green, blue (RGB) color space is most commonly used in computer systems. However, there is a high correlation between channels, and chrominance and luminance are mixed. Since RGB color space is sensitive to noise, it was pointed out that it is not always suitable for color image processing [28],[29]. Hence, we experimentally determine the color space suitable for building extraction. We introduce color spaces where chrominance is separated from luminance in Sections 2.2.2 to 2.2.5.

2.2.2 XYZ

The XYZ color space was devised by the International Commission on Illumination in 1931. It covers all colors visible to the human eye and is categorized as a device-independent color space [28]. The Y channel represents luminance, and the other channels represent chrominance. Conversion from RGB color space to XYZ color space is...
2.2.3 YUV

The YUV color space is used in the analog television standards [28]. The Y channel represents luminance, and the other channels represent chrominance. Conversion from RGB color space to YUV color space is

\[
\begin{bmatrix}
Y \\
U \\
V
\end{bmatrix} =
\begin{bmatrix}
0.299 & 0.587 & 0.114 \\
-0.147 & -0.289 & 0.436 \\
0.615 & -0.515 & -0.100
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}.
\]

(1)

Because RGB, XYZ, and YUV color spaces (from Sections 2.2.1 to 2.2.3) are not close to human perception [30],[31], they are less frequently used in image segmentation tasks [30],[32]. Hence, we introduce HSV and L*a*b* color spaces, which are close to human perception and robust to noise [32] in Sections 2.2.4 and 2.2.5.

2.2.4 HSV

The hue, saturation, value (HSV) color space is designed to be close to the perception of the human eye [31]. It is known as an approximately uniform perceptual color space [29]. The V channel represents luminance, and the other channels represent chrominance. Conversion from RGB color space to HSV color space is

\[
V = \max(R, G, B),
\]

(3)

\[
r = V - \min(R, G, B),
\]

(4)

\[
S = \begin{cases} 
\frac{r}{V} & (V \neq 0), \\
0 & \text{(otherwise)},
\end{cases}
\]

(5)

\[
H' = \begin{cases} 
60(G - B)/r & (V = R), \\
120 + (B - R)/r & (V = G), \\
240 + (R - G)/r & (V = B),
\end{cases}
\]

(6)

\[
H = \begin{cases} 
H' & (H \geq 0), \\
H' + 360 & (H < 0).
\end{cases}
\]

(7)

2.2.5 L*a*b*

L*a*b* color space was derived from XYZ color space. It is more perceptually uniform than HSV color space [33]. The L* channel represents luminance, and the other channels represent chrominance. Conversion from XYZ color space to L*a*b* color space is

\[
\begin{align*}
L' &= 116f(Y/Y_n) - 16, \\
a^* &= 500[f(X/X_n) - (Y/Y_n)], \\
b^* &= 200[f(Y/Y_n) - (Z/Z_n)], \\
f(t) &= \begin{cases} 
13 & (t \leq 0.008856), \\
\frac{t^3}{7.787t + 0.1379} & \text{(otherwise)}.
\end{cases}
\end{align*}
\]

(8)-(11)

Here, \(X_n, Y_n,\) and \(Z_n\) are XYZ values of a reference white point.

2.3 Interactive Segmentation

The conventional method uses the GrowCut algorithm [7], while the proposed method uses the GrabCut algorithm [14]. Both of them segment images by updating the building and background regions repeatedly. This section gives an overview of GrowCut and GrabCut.

2.3.1 GrowCut

The steps of GrowCut, which follows a cellular automata theory, are summarized below.

Step 1: Label \(I_p\), which shows the belonging region of pixel \(p\) (building or background), and strength \(\theta_p\), which shows confidence in the label, are initialized on the basis of the initial seeds.

Step 2: Each pixel \(p\) is checked by using its neighbor pixel \(q\) in the following formula, where \(C_p\) shows the color vector.

\[
1 - \frac{||C_p - C_q||_2}{\max|C|_2}, \quad \theta_q > \theta_p.
\]

(12)

If pixel \(p\) satisfies the above formula, label \(I_p\) and strength \(\theta_p\) are updated as follows:

\[
I_p = I_q, \quad \theta_p = 1 - \frac{||C_p - C_q||_2}{\max|C|_2}, \quad \theta_q.
\]

(13)-(14)

Step 3: Step 2 is iterated until convergence or until a preset iteration limit is reached.

2.3.2 GrabCut

The steps of GrabCut, which follows a graph theory, are summarized below.

Step 1: GMMs for color distribution of the building and background regions (\(g_{\text{build}}\) and \(g_{\text{back}}\), respectively) are estimated on the basis of the initial seeds.

Step 2: Gaussian components of \(g_{\text{build}}\) and \(g_{\text{back}}\) are assigned to pixels.

Step 3: Parameters of \(g_{\text{build}}\) and \(g_{\text{back}}\) are learned.

Step 4: Building and background regions are updated by applying the min-cut/max-flow algorithm [34] to minimize the Gibbs energy function.

Step 5: Steps 2 to 4 are iterated until convergence or until a preset iteration limit is reached.

Although GrowCut and GrabCut are classified as interactive segmentation techniques which require manual input to prepare the initial seeds, the conventional and proposed methods automatically produce the initial seeds on the basis of their assumptions from the tendency of scenery images.

2.4 Conventional Method

The method generates initial seeds for GrowCut [7] on the basis of a binarized image by using the mean adjacent-pixel number [35]. Building regions are extracted as a result of GrowCut being initialized by the initial seeds. The preset iteration limit is not valid for GrowCut in this method. In other words, GrowCut iterates until convergence is reached. The details are described in the reference [6].

3. Proposed Method

Figure 2 shows a flowchart of the proposed method. As mentioned, scenery images tend to have background regions at the top and bottom. The proposed method is designed to extract building regions under the assumption that scenery images, whose main subjects are buildings, have this tendency.

Sections 3.1 to 3.3 explain each step in Fig. 2, and the aforementioned tendency is verified in Section 3.4.
3.1 Color Space Conversion and Color Segmentation

The color space of an input image is converted by using the transformation formula in Section 2.2. The converted color space is used in subsequent processes. A color segmented image is generated by using the clustering method in Section 2.1. Figure 3 (a) shows a color image segmented by applying k-means to the HSV color space of Fig. 1 (a). Each color represents different color clusters.

3.2 Analysis of Color Segmented Image

As mentioned, the background regions tend to be at the top and bottom of scenery images since they are captured by looking forward and upward from the ground. Hence, by analyzing the color segmented image in Section 3.1, color clusters distributed in the upper and lower parts of the image were extracted. The content of each color cluster in the \( N \) rectangular regions \( R_n \) moving toward the center of the image as shown in Fig. 4 is obtained by the following formula:

\[
\begin{align*}
 f^p_n &= \sum_{x=0}^{W-1} \sum_{y=n}^{y_n+H-1} \delta(p_{x,y}, p) \cdot \frac{1}{W \cdot H}, \\
 y_n &= S \cdot n, \quad \delta \text{ is the Kronecker delta,} \\
 W \text{ and } H &\text{ are the width and height of the rectangular regions, } S \text{ is the shift width of } R_n, \\
 y_n &\text{ shows the y-coordinate of the top of } R_n, \text{ and } p_{x,y} \text{ is the cluster number at pixel position } (x,y) \text{ obtained in Section 3.1.}
\end{align*}
\]

Focusing on Fig. 4, \( f^p_n \) of the red cluster (including sky regions) tends to decrease as \( n \) increases, and \( f^p_n \) of the blue cluster (including building regions) tends to increase. The initial seed of the background region, therefore, is obtained on the basis of the cluster \( p \) that satisfies the following conditions:

(I) \( f^p_n \) tends to decrease monotonously as \( n \) increases.

(II) The minimum \( n_{\text{min}} \) at which \( f^p_n \) takes a local minimum value is smaller than \( n_{\text{max}} \); otherwise, \( n_{\text{max}} \) is not found.

The pixels belonging to the cluster that satisfies conditions (I) and (II) for \( 0 \leq y < y_{N-1} \) and \( 0 \leq y < y_{n_{\text{min}}} \), respectively, are considered as candidates for the initial seed of the background.

To extract color clusters distributed in the lower part of an image, the same processing is applied to the image when rotated by 180 degrees.

As a result of analyzing the color segmented image [Fig. 3 (a)], the black regions of Fig. 3 (b) are obtained as candidates for the initial seed of the background.

3.3 Initial Seed Generation and GrabCut

The initial seed of the background is obtained as a result of eroding the candidates for the initial seed of the background (see Section 3.2) \( m \) times in order to remove building regions. The remaining regions are used for the initial seed of the building. The connected components of the initial seed of the background that make contact with the edge of the image are hard-labeled so as not to be changed to building regions by GrabCut since they are more likely to be background regions.

GrabCut, which is initialized by the generated initial seed, is applied. Although GrabCut is categorized as an interactive segmentation technique similar to GrowCut used in the conventional method, we selected GrabCut for the proposed method on the basis of the experimental results demonstrating its superior computational time and accuracy. For example, GrabCut is 5 to 30 times faster than GrowCut without sacrificing the extraction accuracy under the same input conditions on several datasets [36],[37]. Although the reason is not sufficiently discussed in the references, it is possible that GrowCut requires a larger number of iterations to converge than does GrabCut. For examples, while GrowCut requires more than 700 iterations [38], GrabCut requires only 10 or less [39],[40].

The pink and green regions of Fig. 3 (c) are the initial seed of the background and building, respectively. Figure 3 (d) shows...
the building regions extracted by applying GrabCut initialized by the initial seed of Fig. 3 (c).

3.4 Analysis of Scenery Image

The proposed method extracts buildings under the assumption that scenery images whose main subjects are buildings tend to have background regions at the top and bottom. Although this assumption is important for the proposed method, it is not sufficiently verified in reference [15]. Thus, this section supports the assumption by analyzing image areas where building regions are frequently found.

One hundred and six high-quality scenery images captured during sunlight hours from the ZuBuD [16] (hereafter called HQ dataset) and 89 low-quality images captured during night using smartphones (hereafter called LQ dataset) were used for analysis.

Figures 5 (a) and (b) show probabilities of finding building regions in each pixel of the HQ and LQ datasets, respectively. These figures are obtained from the images that indicate building regions and were resized to 256 $\times$ 256 pixels. According to Fig. 5, the said probability decreases in the upper and lower regions in both datasets. Since the assumption is confirmed to be valid for both datasets from different environments, the proposed method is expected to perform well on images whose main subjects are buildings.

4. Experiment

4.1 Experimental Conditions

In our experiment, the HQ and LQ datasets were used for evaluation; their details are given in Section 3.4. Figure 6 shows examples from each dataset. As shown, the LQ dataset contains more noisy, low-contrast images. Ground truth data indicating building regions were prepared manually.

We conducted a comparison experiment to show the effectiveness of the proposed method after a preliminary experiment. In the preliminary experiment, we selected a suitable color space for the proposed method in terms of extraction accuracy. The suitable color space was selected from among the RGB, XYZ, YUV, HSV, and L*a*b* shown in Section 2.2. In the comparison experiment, we evaluated the proposed method in terms of computational time and extraction accuracy. The conventional method, SegNet-Basic, SegNet, and interactive GrabCut were selected for comparison with the proposed method.

Since the conventional method can extract building regions automatically without annotated training data just like the proposed method, the proposed method should be effective in terms of both accuracy and computational time.

SegNet [9], which is one of the representative segmentation networks that uses deep learning, requires training data containing the ground truth. The training cost of SegNet, however, can be decreased since various trained models and training datasets have tended to be available on the Internet for study purposes in recent years. Hence, we consider that the proposed method should improve the extraction accuracy compared with SegNet models, which are available on the Internet or can be trained on a public dataset to show its effectiveness. For this reason, we selected two types of SegNet [SegNet and SegNet-Basic (which is a smaller version of SegNet)] for comparison. The SegNet-Basic model provided by ChainerCV [41] was used for the evaluation. The SegNet model was obtained from transfer learning using pre-trained VGG-16 encoder [42] on ImageNet [43]. Both models were trained with 367 scenery images from CamVid [44].

Interactive GrabCut requires an initial seed generated manually. In this experiment, initial seeds were created by using a bounding box of building regions from ground truth data. Interactive GrabCut accurately extracts building regions since the ideal initial seeds are available. To confirm the best performance of the GrabCut-based method, interactive GrabCut was selected for comparison.

Positive predictive value (PPV), negative predictive value (NPV), and accuracy (ACC) were used to evaluate extraction accuracy.

- True Positive (TP): Total number of correctly determined pixels actually in the building regions.
- False Positive (FP): Total number of incorrectly determined pixels actually in the background regions.
- True Negative (TN): Total number of correctly determined pixels actually in the background regions.
- False Negative (FN): Total number of incorrectly determined pixels actually in the building regions.
- PPV: Ratio of the number of correctly determined pixels to the total number of pixels determined as a building.

$$PPV = \frac{TP}{TP + FP}.$$  (17)
Table 1  Average extraction accuracy (ACC) of the proposed method using different color spaces (values in parentheses show standard deviations).

(a) HQ dataset

|          | RGB (%) | XYZ (%) | YUV (%) | HSV (%)  | L*a*b* (%) |
|----------|---------|---------|---------|----------|-------------|
| K-means  | 85.00 (10.76) | 85.08 (10.59) | 85.39 (9.23) | 85.33 (10.03) | 85.37 (10.07) |
| Mean-shift | 85.77 (9.00) | 84.99 (9.85) | 84.51 (10.18) | 84.57 (11.25) | 84.35 (10.91) |
| GMM      | 84.31 (10.09) | 84.19 (11.24) | 84.97 (9.67) | 85.21 (10.08) | 84.48 (9.50)   |
| VBGMM    | 84.10 (10.37) | 84.43 (9.03) | 84.94 (10.05) | 85.34 (11.01) | 84.78 (10.72) |
| Total    | 84.80 (10.06) | 84.67 (10.19) | 84.95 (9.76)  | 85.11 (10.58) | 84.75 (10.29) |

(b) LQ image

|          | RGB (%) | XYZ (%) | YUV (%) | HSV (%) | L*a*b* (%) |
|----------|---------|---------|---------|---------|-------------|
| K-means  | 78.31 (11.90) | 79.21 (11.60) | 79.76 (10.00) | 81.30 (9.88) | 77.11 (12.92) |
| Mean-shift | 78.10 (11.77) | 78.46 (11.91) | 80.03 (11.37) | 81.77 (10.36) | 77.82 (12.48) |
| GMM      | 78.48 (12.22) | 78.58 (12.13) | 76.62 (13.48) | 81.62 (10.17) | 78.47 (11.54) |
| VBGMM    | 79.74 (12.17) | 78.01 (13.48) | 76.49 (11.16) | 81.67 (10.82) | 77.67 (11.47) |
| Total    | 78.66 (11.98) | 78.56 (12.26) | 78.23 (11.64) | 81.44 (10.28) | 77.77 (12.08) |

Table 2  Average computational time for HQ and LQ dataset.

|          | Conventional method (s) | Proposed method |
|----------|-------------------------|-----------------|
|          | k-means (s) | Mean-shift (s) | GMM (s) | VBGMM (s) | SegNet-Basic (s) | SegNet (s) |
| HQ       | 27.21       | 19.76          | 35.59   | 7.84      | 33.06          | 13.20     | 22.72 |
| LQ       | 40.73       | 38.33          | 87.96   | 53.71     | 150.44         | 13.27     | 23.15 |

- NPV: Ratio of the number of correctly determined pixels to the total number of pixels determined as the background.

\[
NPV = \frac{TN}{TN + FN}. \tag{18}
\]

- ACC: Ratio of the number of correctly determined pixels to the total number of pixels.

\[
ACC = \frac{TP + TN}{TP + FP + TN + FN}. \tag{19}
\]

The mean value of the computational time per image was used for evaluating the processing speed of each method. Each method was written in Python by using scientific libraries (Numpy, scikit-learn, OpenCV, Chainer, and ChaineCV). GrowCut and GrabCut were implemented with C++ and called by Python. SegNet was performed on a CPU just like the other methods.

4.2 Preliminary Experiment Result

Table 1 shows the extraction accuracy of the proposed method using different color spaces. The parameters were experimentally tuned on the basis of the accuracy. Comparing the mean extraction accuracy of each color space, we can confirm that HSV color space had higher accuracy than the other color spaces for both the HQ and LQ datasets [Tables 1 (a) and (b), respectively].

Compared with our previous method [15], which uses only RGB color space, the proposed method improved the extraction accuracy by 0.31% and 2.78% for the HQ and LQ datasets, respectively.

The differences between extraction accuracies were evaluated with Tukey’s test and the Steel-Dwass test, which are categorized as parametric and non-parametric multiple comparison tests, respectively, at a significance level 0.05. The extraction accuracy of HSV color space was found to be significantly different from the other color spaces for the LQ dataset (the observed p-value was less than 2.3%). From the above, HSV color space is considered to be suitable for the proposed method, especially for the LQ dataset.

Figure 7 compares the results of building extraction by the previous method based on RGB color space and the proposed method based on HSV color space. The previous method has a high rate of buildings erroneously determined as the background. This result suggests that the color difference between the background and the buildings may be difficult to distinguish in RGB color space due to the high correlation between channels, unlike in HSV color space.

4.3 Comparison Experiment Result

4.3.1 Computational time

Table 2 shows the computational time of each method. To be faster than the conventional method, the proposed method should select k-means as the color clustering method, although, with GMM, the computational time of the proposed method is lower for the HQ dataset.

Compared with the conventional method, the proposed method using k-means could reduce the computational time by 27.38% and 5.89% for the HQ and LQ datasets, respectively.

Here, we confirm the extraction accuracy of the proposed method using k-means compared with that using the other color clustering methods by focusing on the fifth column of Tables 1 (a) and (b). The extraction accuracy of the proposed...
Table 3  Average extraction accuracy (ACC) of each method (values in parentheses show standard deviations).

(a) HQ dataset

|          | Conventional method (%) | Proposed method (%) | SegNet-Basic (%) | SegNet (%) | Interactive GrabCut (%) |
|----------|-------------------------|---------------------|------------------|------------|-------------------------|
| PPV      | 73.97 (17.17)           | 84.92 (13.75)       | 89.01 (10.13)    | 91.74 (9.05) | 87.33 (8.67)           |
| NPV      | 53.95 (28.93)           | 86.80 (21.81)       | 61.18 (22.91)    | 70.76 (18.88) | 91.57 (9.99)           |
| ACC      | 70.48 (15.52)           | 85.33 (10.03)       | 77.98 (8.95)     | 84.14 (7.32) | 88.31 (7.17)           |

(b) LQ dataset

|          | Conventional method (%) | Proposed method (%) | SegNet-Basic (%) | SegNet (%) | Interactive GrabCut (%) |
|----------|-------------------------|---------------------|------------------|------------|-------------------------|
| PPV      | 66.28 (16.48)           | 83.60 (12.32)       | 78.97 (16.61)    | 87.23 (9.99) | 88.61 (8.68)           |
| NPV      | 51.12 (34.24)           | 77.19 (22.10)       | 49.50 (19.95)    | 54.35 (19.61) | 81.26 (17.16)           |
| ACC      | 62.36 (12.18)           | 81.30 (9.89)        | 61.19 (14.49)    | 67.11 (14.41) | 85.79 (9.79)           |

Fig. 8  Results of building extraction with each method.

method using k-means was not the highest for both datasets. Significant differences, however, were not be found as a result of the multiple comparison tests in Section 4.2 (the observed p-value was more than 85%).

On the basis of these experimental results, the proposed method hereafter uses k-means as the color clustering method.

4.3.2  Extraction accuracy

Table 3 shows a summary of the extraction accuracy for each method. Compared with the conventional method, the proposed method improved all three evaluation indexes for both of the datasets. It improved ACC (comprehensive extraction accuracy measurement) by 14.85% and 18.94% for both the HQ and LQ datasets, respectively.

Compared with two types of SegNet models, the proposed method increased ACC by 1.19% or more for both of the datasets, whereas the conventional method decreased ACC by 7.50% or more for the HQ dataset. This shows that the proposed method is superior to deep-learning-based segmentation models available on the Internet or learned on a public dataset such as CamVid.

Compared with interactive GrabCut, the proposed method decreased ACC by 2.98% and 4.49% for the HQ and LQ datasets, respectively. This implies that it has room for improvement in terms of the process of generating the initial seed for GrabCut. The proposed method, however, is effective in realizing automation compared with interactive GrabCut.

Here, the results of the multiple comparison tests (see Section 4.2) related to the proposed method are presented. Tukey’s and the Steel-Dwass tests showed that the extraction accuracy (ACC) of the proposed method was significantly different from that of the conventional method and SegNet-Basic for both of the datasets (the observed p-value was less than 0.001%). The extraction accuracy of the proposed method was also significantly different from that of SegNet on the LQ dataset in both tests (the observed p-value was less than 0.001%). As for the results between the proposed method and interactive GrabCut, a significant difference was confirmed by the Steel-Dwass test for the LQ dataset (the observed p-value was less than 0.4%).

Figure 8 shows two examples of building regions extracted by each method. The conventional method had a high rate of wrongly determining the sky and road as building regions compared with the other methods. This tendency was frequently observed in the other images and decreased the extraction accuracy as mentioned in Section 1. In comparison, the proposed method can extract building regions accurately like interactive GrabCut.

5. Discussion

In Section 4, the proposed method based on the color segmentation by k-means and GrabCut with the HSV color space was found to be effective in terms of both extraction accuracy and computational time compared with the conventional method. In this section, a detailed discussion is presented on the basis of the experimental results.

5.1  Noise Robustness of HSV Color Space

The quality of the several images used in our experiment decreased because of noise and loss of contrast (see Fig. 9). Noisy, low-contrast images usually degrade the performance of image processing tasks [45]. Since the LQ dataset contains relatively noisier, lower-contrast images compared with the HQ dataset, the extraction accuracy of the LQ dataset tended to decrease (see Table 1).

Here, looking at Table 1 (b), the extraction accuracy of HSV color space was higher than that of the other color spaces. HSV color space is known to be suitable for dealing with noisy images compared with other color spaces. For example, the noise robustness of HSV color space for image segmentation is illus-
The conventional and proposed methods are roughly divided into two processes: initial seed generation and segmentation.

Fig. 9 Example of noisy, low-contrast image.

Fig. 10 Example of mixed cluster ((b) and (c) are color segmented image by k-means).

trated in references [19],[32]. This merit may have improved the extraction accuracy better than the other color spaces in our experiment.

The noise robustness of HSV color space contributed to the proposed method significantly improving the extraction accuracy by 1.55% on average compared with our previous method (using RGB color space) [15]. In particular, for the LQ dataset, the proposed method showed more of an improvement (2.78%).

For more accurate building extraction, high-performance image sensors and imaging techniques that can enhance the quality of noisy, low-contrast images should be developed, although the proposed method needs to be refined to improve noise robustness.

5.2 Number of Generated Color Clusters

Because color differences between the building and background regions were quite small in the LQ dataset due to its low contrast, mixed clusters containing both building and background regions [e.g., sky and a part of building cluster in Fig. 10(b)] tended to be generated more frequently. Although the proposed method permits the presence of mixed clusters to some extent, it is better for the number and size of them to be reduced to maintain the extraction accuracy.

Regarding the determined parameters of the proposed method, the number of clusters generated for the LQ dataset was increased compared with that for the HQ dataset. This contributed to reducing the number and size of mixed clusters by increasing the number of generated clusters as shown in Fig. 10(c), although over-clustering occurred.

5.3 Analysis of Computational Time

The proposed method reduced computational time more than 5% compared with the conventional method (see Section 4.3.1). In this section, we, therefore, analyze the computational time of the proposed method by comparing it with that of the conventional method.

The conventional and proposed methods are roughly divided into two processes: initial seed generation and segmentation, which is performed by GrowCut in the conventional method or by GrabCut in the proposed method. Table 4 shows the average computational time of each process.

The computational time of the initial seed generation process of the proposed method was higher than that of the conventional method. This is because color segmentation by k-means in initial seed generation requires a larger amount of computational time. We confirmed that the computational time of k-means accounts for more than 97% of that of the initial seed generation process. The computational time of k-means for the LQ dataset was increased 1.76 times compared with that for the HQ dataset. This is because the number of color clusters generated from k-means in the LQ dataset was increased compared with that in the HQ dataset as mentioned in Section 5.2. The computational time of k-means increases according to the number of generated clusters [24].

Compared with the conventional method, the proposed method reduced the computational time for the segmentation process, as GrabCut is faster than GrowCut according to references [36],[37].

Focusing on the proposed method, the computational time of the segmentation process for the LQ dataset increased 2.71 times compared with that for the HQ dataset. This may be influenced by the noise of the low-contrast images included in the LQ dataset since high-intensity noise increases the computational time of the EM algorithm used in GrabCut [46].

From the above, we clarified that k-means and GrabCut included in the proposed method should be accelerated under noisy environment to create a much faster method.

6. Conclusion

In this paper, we proposed a method for automatically extracting buildings from scenery images. The proposed method is designed to extract building regions in scenery images under the assumption that the background regions tend to be found in the upper and lower parts if buildings are the main subjects in such images. Based on experimental results, color segmentation by k-means and GrabCut with the HSV color space were selected for the proposed method to improve the extraction accuracy and computational time.

To confirm the effectiveness of the proposed method, we evaluated its extraction accuracy and computational time by using 106 high-quality scenery images (HQ dataset) and 89 low-quality ones (LQ dataset). The LQ dataset, which contained noisier, lower-contrast images captured at night, was prepared so as to enable the method to extract buildings accurately in various environments.

Compared with the conventional method, the proposed method significantly improved the extraction accuracy by 14.85% and 18.94% for the HQ and LQ datasets, respectively. It also reduced the consumption time by 27.38% and 5.89% for the HQ and LQ datasets, respectively. This result shows that the proposed method is effective in terms of both extraction ac-
curacy and computational time.

The extraction accuracy was improved by 1.19% or more compared with deep-learning-based segmentation models which are available on the Internet or trained on a public dataset.

Confirming the extraction accuracy of the proposed method by using different color spaces, HSV color space significantly improved the extraction accuracy by more than 0.16% and 2.78% compared with the other color spaces for the HQ and LQ datasets, respectively. This result shows the noise robustness of HSV color space confirmed in other studies since HSV color space was more effective for the LQ dataset.

Compared with iterative GrabCut, which uses initial seeds generated manually, the proposed method worsened the extraction accuracy by 2.98% and 4.49%, respectively. This suggests that the extraction accuracy of the proposed method could be improved by modifying the process of generating the initial seed for GrabCut. Further studies are needed in order to improve the extraction accuracy.

References

[1] P.D. Groves: Shadow matching: A new GNSS positioning technique for urban canyons, Journal of Navigation, Vol. 64, No. 3, pp. 417–430, 2011.

[2] Z. Jiang and P.D. Groves: GNSS NLOS and multipath error mitigation using advanced multi-constellation consistency checking with height aiding, Proc. 25th International Technical Meeting of the Satellite Division of the Institute of Navigation, pp. 79–88, 2012.

[3] W. Zhang and J. Kosecka: Localization based on building recognition, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 21–28, 2005.

[4] Y.C. Chung, H.X. Tony, and Z. He: Building recognition using sketch-based representations and spectral graph matching, Proc. IEEE International Conference on Computer Vision, pp. 2014–2020, 2009.

[5] J. Åhlen and S. Seipei: Knowledge based single building extraction and recognition, Proc. WSEAS International Conference on Computer Engineering and Applications, pp. 29–35, 2014.

[6] D. Ueno, H. Yoshida, and Y. Iiguni: Automated GrowCut with multilevel seed strength value for building image, Transactions of the Institute of Systems, Control and Information Engineers, Vol. 29, No. 6, pp. 266–274, 2016 (in Japanese).

[7] V. Vezhnevets and V. Konouchine: GrowCut: Interactive multi-label N-D image segmentation by cellular automata, Proc. Graphicon, pp. 150–156, 2005.

[8] H. Noh, S. Hong, and B. Han: Learning deconvolution network for semantic segmentation, Proc. IEEE International Conference on Computer Vision, pp. 1520–1528, 2015.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla: SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 12, pp. 2481–2495, 2017.

[10] A. Fond, M.O. Berger, and G. Simon: Joint facade registration and segmentation for urban localization, arXiv preprint arXiv:1811.10048, 2018.

[11] D. Choi, H. Cho, and W. Rhee: On the difficulty of DNN hyper-parameter optimization using learning curve prediction, Proc. 2018 IEEE Region 10 Conference, pp. 651–656, 2018.

[12] K. Weiss, T.M. Khoshgoftaar, and D. Wang: A survey of transfer learning, Journal of Big Data, Vol. 3, No. 1, p. 9, 2016.

[13] C. Shorten and T.M. Khoshgoftaar: A survey on image data augmentation for deep learning, Journal of Big Data, Vol. 6, No. 1, p. 60, 2019.

[14] C. Rother, V. Kolmogorov, and A. Blake: GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics, Vol. 23, No. 3, pp. 309–314, 2004.

[15] T. Futagami and N. Hayasaka: Automatic extraction of building regions by using color clustering, Proc. SICE Annual Conference 2019, pp. 415–419, 2019.

[16] H. Shao, T. Svoboda, and L. Van Gool: Zubud - Zurich building database for image based recognition, ETH Zurich, Technical Report, 260, 2003.

[17] D. Khattab, H.M. Ebied, and A.S. Hussein: Color image segmentation based on different color space model using automatic GrabCut, The Scientific World Journal, pp. 126025:1–126025:10, 2014.

[18] S. Chebbou and H.F. Merouani: Comparative study of clustering based colour image segmentation techniques, Proc. 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems, pp. 839–844, 2015.

[19] D.J. Bora: Importance of image enhancement techniques in color image segmentation: A comprehensive and comparative study, International Journal of Science and Research, Vol. 15, No. 1, pp. 115–131, 2017.

[20] J.B. MacQueen: Some methods for classification and analysis of multivariate observations, Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–291, 1967.

[21] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A.Y. Zomaya, S. Foufou, and A. Bouras: A survey of clustering algorithms for big data: Taxonomy and empirical analysis, IEEE Transactions on Emerging Topics in Computing, Vol. 2, No. 3, pp. 267–279, 2014.

[22] D. Comaniciu and P. Meer: Mean shift: A robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, pp. 603–619, 2002.

[23] M.G. Omran, A.P. Engelbrecht, and A. Salman: An overview of clustering methods, Intelligent Data Analysis, Vol. 11, No. 6, pp. 583–605, 2007.

[24] D. Xu and Y. Tian: A comprehensive survey of clustering algorithms, Annals of Data Science, Vol. 2, No. 2, pp. 165–193, 2015.

[25] C.E. Rasmussen: The infinite Gaussian mixture model, Advances in Neural Information Processing Systems, Vol. 12, pp. 554–560, 2000.

[26] C.M. Bishop: Pattern Recognition and Machine Learning, pp. 461–517, Springer-Verlag New York, 2006.

[27] A.Y. Ng, M.I. Jordan, and Y. Weiss: On spectral clustering: analysis and an algorithm, Advanced in Neural Information Processing Systems, Vol. 14, No. 2, pp. 849–856, 2002.

[28] M.H. Asmare, V.S. Asirvadam, and L. Iznita: Color space selection for color image enhancement applications, Proc. 2009 International Conference on Signal Acquisition and Processing, pp. 208–212, 2009.

[29] V. Chernov, J. Alander, and V. Bochko: Integer-based accurate conversion between RGB and HSV color spaces, Computers & Electrical Engineering, Vol. 46, No. 1, pp. 328–337, 2015.

[30] T.W. Chen, Y.L. Chen, and S.Y. Chien: Fast image segmentation based on K-means clustering with histogram in HSV color space, Proc. 2008 IEEE 10th Workshop on Multimedia Signal Processing, pp. 322–325, 2008.

[31] K.N. Plataniotis and A.N. Venetsanopoulos: Color Image Processing and Applications, pp. 1–45, Springer-Verlag Berlin Heidelberg, 2000.

[32] J. Bora, A.K. Gupta, and F.A. Khan: Comparing the performance of L*A*B* and HSV color spaces with respect to color image segmentation, International Journal of Emerging Technology and Advanced Engineering, Vol. 5, No. 2, pp. 192–203, 2015.

[33] K.V. Velde: Multi-scale color image enhancement, Proc.
1999 International Conference on Image Processing, Vol. 3, pp. 584–587, 1999.

[34] Y. Boykov and V. Kolmogorov: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1124–1137, 2004.

[35] K. Sasakawa, S. Kuroda, and S. Ikebata: A method for threshold selection in binary images using mean adjacent-pixel number, *Systems and Computers in Japan*, Vol. 22, No. 3, pp. 66–73, 1991.

[36] X. Li, K. Liu, and Y. Dong: Superpixel-based foreground extraction with fast adaptive trimaps, *IEEE Transactions on Cybernetics*, Vol. 48, No. 9, pp. 2609–2619, 2017.

[37] X. Song, L. Zhou, Z. Li, J. Chen, B. Yan, and L. Zeng: Interactive image segmentation based on hierarchical superpixels initialization and region merging, *Proc. 2014 7th International Congress on Image and Signal Processing*, pp. 410–414, 2014.

[38] S. Bhowal and B.C. Dhara: Segmentation of similar color objects using modified GrowCut method, *International Journal of Advanced Computer Research*, Vol. 3, No. 3, pp. 36–41, 2013.

[39] D. Corrigan, S. Robinson, and A. Kokaram: Video matting using motion extended GrabCut, *Proc. IET 5th European Conference on Visual Media Production*, pp. 1–9, 2008.

[40] P. Shanmugavadivu and G. Thenmozhi: Detection of microcalcification in mammogram images using semi-automated texture based GrabCut Segmentation, *2012 International Conference on Emerging Trends in Science, Engineering and Technology (INCOSET)*, pp. 198–203, 2012.

[41] Y. Niitani, T. Ogawa, S. Saito, and M. Saito: ChainerCV: A library for deep learning in computer vision, *Proc. 25th ACM International Conference on Multimedia*, pp. 1217–1220, 2017.

[42] K. Simonyan and A. Zisserman: Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[43] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei: ImageNet: A large-scale hierarchical image database, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[44] G. Brostow, J. Fauqueur, and R. Cipolla: Semantic object classes in video: A high-definition ground truth database, *Pattern Recognition Letters*, Vol. 30, No. 2, pp. 88–97, 2009.

[45] K. Huang, L. Wang, T. Tan, and S. Maybank: A real-time object detecting and tracking system for outdoor night surveillance, *Pattern Recognition*, Vol. 41, No. 1, pp. 432–444, 2008.

[46] O. Osonde and B. Kosko: The noisy expectation-maximization algorithm for multiplicative noise injection, *Fluctuation and Noise Letters*, Vol. 15, No. 1, pp. 1650007:1–1650007:23, 2016.

---

**Takao Onoye**

He received the B.E. and M.E. degrees in Electronic Engineering, and Dr.Eng. degree in Information Systems Engineering, all from Osaka University, Japan, in 1991, 1993, and 1997, respectively. He is currently a professor in the Department of Information Systems Engineering, Osaka University. His research interests include media-centric low-power architecture and its SoC implementation. He is a member of IEEE, IEICE, IPSJ, and ITE-J.

---

**Takuya Futagami (Member)**

He received his B.S. and M.S. degrees from Osaka University, Japan, in 2013 and 2015. He has been a visiting researcher at Osaka Electro-Communication University since 2018. In 2019, he joined Osaka University, where he is currently a Ph.D. student. His current research interests are image processing and image recognition. He is a member of ISCIE, IEICE, and IEEJ.

**Noboru Hayesaka**

He received his B.S., M.S., and Ph.D. degrees from Hokkaido University in 2002, 2004, and 2007. He is currently an associate professor at Osaka Electro-Communication University. His current research interests are speech processing and speech recognition. He is a member of ASJ, IEICE, and IEEE.