

Button Segmentation from an Elevator Inside Door Image

Yu-Ching Hsu and Yung-Sheng Chen *

Abstract—Computer vision on the segmentation of buttons from an elevator inside door (EID) image is studied. Due to the reflection phenomenon and the similar color between buttons and the door panel consisting of the buttons in the EID image, it is not easy to deal with the button segmentation. To overcome this problem, based on the edge information of the EID image a kernel function called projection-and-checking (PAC) is presented in this paper for the button segmentation, which will be valuable for the robot vision application. Our experiments confirm the feasibility of the proposed approach.

Keywords: computer vision, edge information, elevator inside door (EID), image segmentation, projection and checking (PAC)

1 Introduction

Computer vision plays an important role for a robotic system, in particular considering the interaction between a robot and the environment [1]. Except for the study on the interaction between robot and real world, it is also very interesting for exploring that between robot and digital world, such as the CUBot developed by Chen and Lin [2]. The CUBot possesses the abilities of perceiving the information on the computer screen [3] and manipulating the compute mouse to operate the computer like a human being. The CUBot is very suitable for investigating the eye-hand coordination mechanism. In addition, the recognition of keyboard was also studied for its further integration into the CUBot in the near future [4]. Along such a research direction, another interesting topic on the button recognition of the elevator door control is opened here and presented in this paper. Due to the prompt progress of robot technology, the various robots are coming into our daily life. Consider a scenario that a mobile robot wants to get an elevator as we usually do in our daily life, one of most important key functions is that the buttons of elevator should be perceived by the robot vision. The elevator inside door (EID) image containing buttons is therefore studied. Segmentation of buttons from the EID image is a necessary preprocessing for a human-like robot to operate the elevator and thus

is the goal of this paper to propose an automatic method for dealing with this topic.

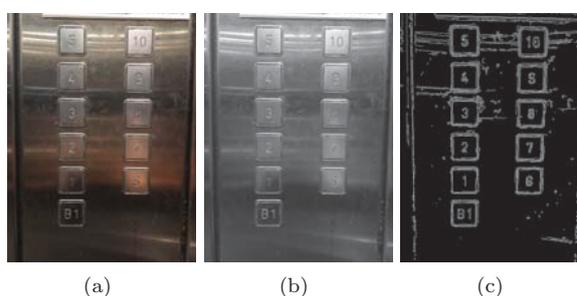


Figure 1: Elevator inside door (EID) image with (a) original (color), (b) gray, and (c) edge domain.

Given an EID-image as shown in Figure 1(a), some properties can be observed as follows. The color of the door panel and that of buttons are very similar and not easily distinguished. In addition, there is a serious reflection over the EID-image. Therefore, the traditional method, e.g., SIFT [5], will not be helpful for such a situation of unobvious features. Note here that the SIFT technology has been successfully applied for the keyboard recognition [4], where the keyboard image possesses obvious features on each key. Except for the SIFT consideration, it is also possible that by means of a thinning method [6] to extract the stroke information of characters inside button and then recognize the button. However, before performing such a scheme a good thresholding preprocessing should be available but it is not easy due to the mentioned issues of reflection phenomenon and similar color. Fortunately, the edge information appearing in the EID image provide us a cue for designing a feasible approach of locating the buttons, which will be presented in this paper.

The rest of this paper is organized as follows. The proposed method is presented in Section 2, which includes our kernel function PAC (projection-and-checking), the segmentation of button columns from an EID image, the segmentation of buttons from a button column, as well as a refining process for locating the characters inside buttons. The result and conclusion are finally drawn in Section 3.

*Y.-C. Hsu and Y.-S. Chen are with the Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan, ROC. (phone: 886-3-4638800; fax: 886-3-4639355; e-mail: ruby841009@gmail.com; ceyschen@saturn.yzu.edu.tw).

2 Proposed Method

As mentioned previously, there exist the inevitable problems of reflection phenomenon and similar color in the EID image. By observing the original (color) EID image given in Figure 1(a), the reflection phenomenon is very serious and the metal color makes the button and its symbol difficultly differentiated. Fortunately, the edge information on the EID image provides us a cue for designing our approach. Therefore, before presenting our approach the original EID image is first converted into a gray-scale one as shown in Figure 1(b) and then further using the Canny edge detection method [7] to extract the wanted edge information as shown in Figure 1(c), which is called E-image latter for the sake of presenting our method.

Let $e(x, y)$ be a bi-level pixel belonging to the E-image with the size of $W \times H$ pixels, x and y be the coordinate along the horizontal and vertical direction, respectively. The origin $(0, 0)$ is located at the left-top corner of the E-image. We then have

$$e(x, y) = \begin{cases} 1, & \text{edge pixel (called 1-pixel),} \\ 0, & \text{otherwise (called 0-pixel).} \end{cases} \quad (1)$$

Refer to the image shown in Figure 1(c), even some unwanted edge information appear in the E-image, the appearance of edge information for each button region can be used for the button segmentation. The more the collected edge information, the higher the possibility of being a button region. In order to reduce the noisy pixels in the edge image and make the appearance information more enough, a morphological operation of three times erosions and dilations with 3×3 structure element is applied for the E-image. Along the current illustration, the obtained morphology result is named as M-image as shown in Figure 2, in which the pixel is denoted by $m(x, y)$. The adopted idea of developing our approach is to project the accumulation of 1-pixel in the M-image onto horizontal and vertical axis iteratively followed with some checking process to identify a useful range ($R_x = [x_1 : x_2]$, or $R_y = [y_1 : y_2]$), which can be used to form a region-of-interest (ROI) image cropped from the M-image and denoted as $ROI(R_x, R_y) = ROI([x_1 : x_2], [y_1 : y_2])$. For example, in this way, the original M-image can be represented as $ROI([0 : W - 1], [0 : H - 1])$. Thus the kernel function in this study is named as projection-and-checking (PAC for short) and may be formulated as follows.

PAC: projection-and-checking

Input an ROI-image.

Output some useful ranges.

1. Given an image with $ROI([x_1 : x_2], [y_1 : y_2])$.
2. Perform projection process:

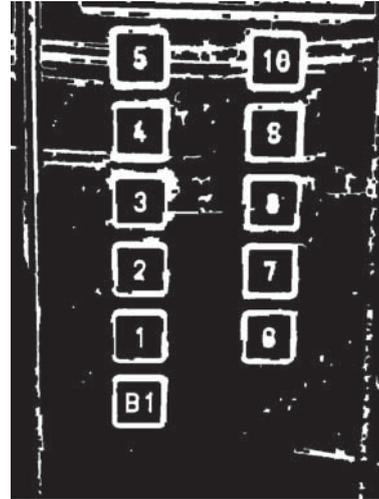


Figure 2: M-image obtained by performing the morphological operations onto the E-image as given in Figure 1(c).

- (a) If vertical projection along x-axis (P_x) is considered, we have the projection array $P_x(i), i = x_1, \dots, x_2$.
 - (b) If horizontal projection along y-axis (P_y) is considered, we have the projection array $P_y(i), i = y_1, \dots, y_2$.
3. Perform checking process:
- (a) If checking process (C_x) is performed on the P_x , we obtain N_x wanted ranges, $R_x(i), i = 1, \dots, N_x$.
 - (b) If checking process (C_y) is performed on the P_y , we obtain N_y wanted ranges, $R_y(i), i = 1, \dots, N_y$.

The PAC function can be further denoted as PAC_x (including P_x and C_x) and PAC_y (including P_y and C_y) depending on the process performed along x-axis or y-axis, respectively.

2.1 Segmentation of button columns

The EID-image is usually composed of several buttons arranged by multiple columns as the two columns shown in Figure 1(a). Therefore, the region of button column should be located first before segmenting the buttons. In this step for our PAC processing, the input ROI-image (ROI_M) is the original M-image as given in Figure 2. Since button columns are considered here, only the process PAC_x will be performed in this stage. By performing P_x , the projection plot can be obtained as depicted in Figure 3(a). From this plot, it is obvious that two main "black" ranges reflects the two button columns, which can be extracted by the following checking process C_x ,

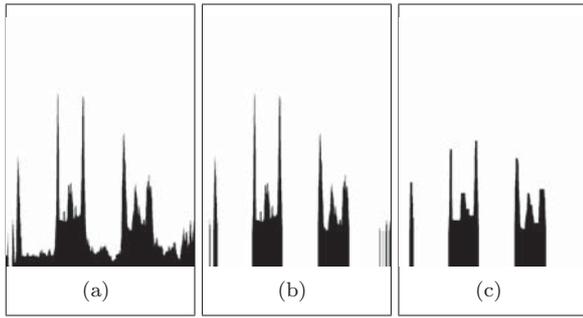


Figure 3: Projection results of performing P_x for the M-image given in Figure 2: (a) original, (b) thresholded, and (c) filtered result.

namely, thresholding, median filtering, and range detection.

Thresholding: Let \bar{p}_x and p_{min} be respectively the mean and the non-zero minimum of the $P_x(i), i = 0, \dots, W - 1$. The thresholding process is designated as

$$P_x(i) = \begin{cases} 0, & \text{if } P_x(i) < (\bar{p}_x + p_{min})/2, \\ P_x(i), & \text{otherwise.} \end{cases} \quad (2)$$

Median filtering: After performing this thresholding scheme onto the plot in Figure 3(a), we can obtain some separated ranges but some unwanted “noise” information still included as the plot shown in Figure 3(b). Therefore a median filtering is further applied onto it and a more clear plot can be obtained as shown in Figure 3(c), which is suitable for the range detection.

Range detection and button column segmentation: By observing the plot in Figure 3(c), we can find some, say N_x , ranges with “black” zones, where their widths are denoted as $W_x(i), i = 1, \dots, N_x$. Let \bar{w}_x be the mean of the $W_x(i), i = 1, \dots, N_x$. That range having its width greater than $\bar{w}_x/2$ will be remained and used to locate the corresponding button column; otherwise are removed. In the current illustration, two wanted range $R_x(1)$ and $R_x(2)$ are found as depicted in Figure 4(a), where the number N_x is reduced to 2. Based on these two ranges, the two button columns can be cropped from the M-image as shown in Figure 4(b) and 4(c), respectively.

2.2 Segmentation of buttons

After locating the button columns, we can obtain the corresponding ROI-image, say ROI_{BC}^1 and ROI_{BC}^2 for the current illustration as shown in Figure 4(b) and 4(c). It can be found that all buttons are confined in these ROIs. Therefore, the same procedure described in Section 2.1 can be applied onto these ROIs one by one for locating the wanted buttons. The projection direction is changed from vertical into horizontal one. That is, the PAC_y including P_y and C_y functions will be performed

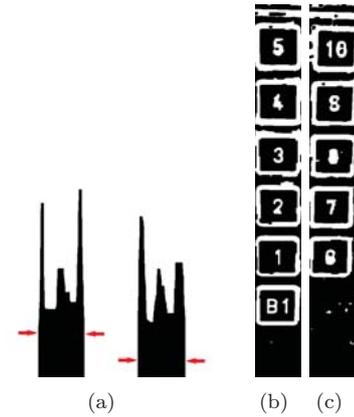


Figure 4: (a) Two detected ranges. (b) and (c) show the cropped button columns (ROI_{BC}^1 and ROI_{BC}^2) from the M-image.

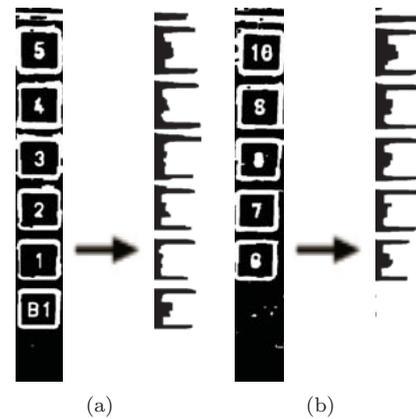


Figure 5: Projection results of performing P_y for the (a) ROI_{BC}^1 and (b) ROI_{BC}^2 .

on the $ROI_{BC}^i, \forall i$. By performing the PAC_y , the corresponding projection plots can be obtained as depicted in Figure 5. For ROI_{BC}^1 , we obtain $N_y^1 = 6$ wanted ranges, i.e., $R_y^1(j), j = 1, \dots, N_y^1$. For ROI_{BC}^2 , we have $N_y^2 = 5$ wanted ranges, i.e., $R_y^2(j), j = 1, \dots, N_y^2$. Thus all the buttons, $ROI_B^k, k = 1, \dots, \sum_{i=1}^{N_x} N_y^i$, can be generally located with the following loops.

Locating all buttons

```

k = 1
For i = 1 to N_x
    For j = 1 to N_y^i
        Output ROI_B^k(R_x(i), R_y^i(j))
    k ++
End For
End For
    
```

Because the character on the button will be finally used for robot vision, its region is further extracted in this study. By observing the projection array for an ROI_B^k , for example $k = 1$, the character range in vertical direction can be easily confined with a further checking for the “red markers” shown in Figure 6(a). Such a checking process (called **y-Updating 1**) is described as follows.

y-Updating 1

1. Given the ROI_B^k with the range $R_x = [x_1 : x_2]$ and $R_y = [y_1 : y_2]$
2. Find the top valley $P_y(y^{top})$ and the bottom valley $P_y(y^{bottom})$
3. By scanning y from y^{top} to y^{bottom} , if the case $P_y(y - 1) < P_y(y)$ AND $|P_y(y - 1) - P_y(y)| > TH$ is first met, then update $y_1 = y$ and stop scanning.
4. By scanning y from y^{bottom} to y^{top} , if the case $P_y(y - 1) > P_y(y)$ AND $|P_y(y - 1) - P_y(y)| > TH$ is first met, then update $y_2 = y - 1$ and stop scanning.

In this study, the $TH = 2$ is selected for experiments. After performing the **y-Updating 1** for all (ROI_B) s, all the wanted buttons can be roughly located as shown in Figure 6(b), here the original EID-image is used for reference.

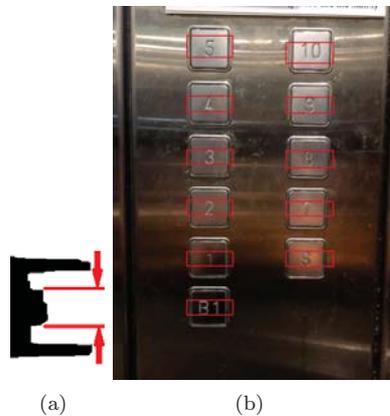


Figure 6: (a) $R_y^1(1)$ of ROI_B^1 modified with a further checking process. (b) All the wanted buttons are roughly located.

2.3 Refining process

By observing the results given in Figure 6(b), it seems that the range of each ROI_B should be further refined in order to locate the wanted characters more close. With the similar way of **y-Updating 1** presented privously, the PAC_x with **x-Updating** and PAC_y with **y-Updating 2** are further applied sequently for refining R_x and R_y of each ROI_B . The former is used to remove the highlighted side projections (which are formed by the vertical projections of button’s frame), whereas the latter

is used to ignore the null side projections (note here that the horizontal projections of button’s frame have been ignored in the process of **y-Updating 1**). The final results of locating the characters inside buttons are shown in Figure 7, which demonstrates the feasibility of the proposed method. As a result, based on the processing steps presented before our approach can be summarized as the flowchart given in Figure 8.



Figure 7: Results of locating the characters inside buttons.

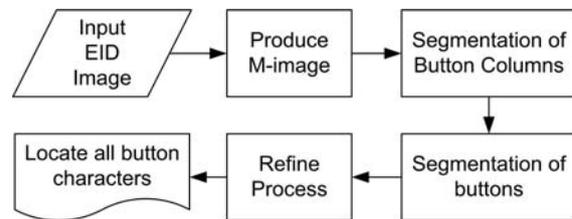


Figure 8: Flowchart of the proposed method.

3 Result and Conclusion

So far we have presented an automatic approach to segment the buttons from the so-called EID image. To facilitate the further character recognition, the range of located button has also been refined to that of characters inside button. Figure 9 show the other two results of locating the characters inside buttons. Even the experiments have demonstrated the feasibility of the proposed approach, there are some efforts that can be improved further. For example, more types of buttons arranged in an EID-image should be investigated for improving the current method more robust to various cases. The recognition mechanism or machine learning may be embedded into the system for accurately identifying the character of a button and thus make the robot vision to be more practicable. It is expected that a home-care mobile robot

can take the elevator from one floor to the other for good service in the near future.

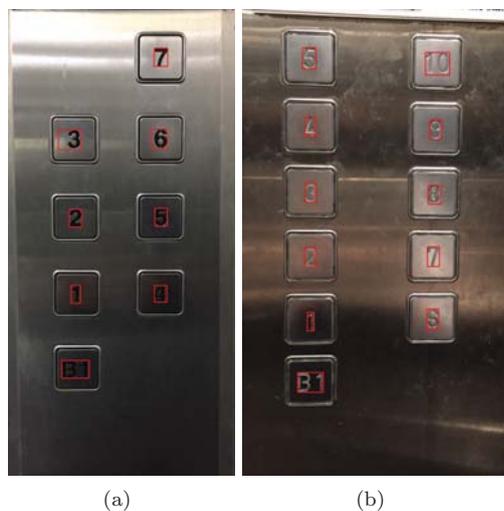


Figure 9: Other two results of locating the characters inside buttons.

References

- [1] S. Chen, Y. Li, and N. Kwok, "Active vision in robotic systems: A survey of recent developments," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343-1377, 2011.
- [2] Y.-S. Chen and K.-L. Lin, "CUBot: Computer vision on the eye-hand coordination with a computer-using robot and its implementation," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 4, 1855005 (25 pages), 2018.
- [3] Y.-S. Chen and K.-L. Lin, "Screen image segmentation and correction for a computer display," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 4, 1454002 (21 pages), 2014.
- [4] M.-T. Chao and Y.-S. Chen, "Keyboard recognition from scale-invariant feature transform," *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, Taipei, Taiwan, pp. 207-208, 2017.
- [5] D. G. Lowe, "Object recognition from local scale-invariant features," *IEEE International Conference on Computer vision*, vol. 2, pp. 1150-1157, 1999.
- [6] Y.-S. Chen and M.-T. Chao, "Pattern reconstructability in fully parallel thinning," *Journal of Imaging*, vol. 3, no. 3, 29 (23 pages), 2017.
- [7] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.