Business Card Analysis

Dominic Landolf
Team member: Sara Vestergren
The Hong Kong University of Science and Technology
djlandolf@connect.ust.hk

Abstract

This report presents a way to analyse the content on a business card in three main steps after pre-processing the image: text detection, character segmentation and optical character recognition. Each part is performed by only using simple methods such as erosion, dilation and vertical projection. With the presented methods, the program is able to successfully highlight the areas of contact information as well as the e-mail address and the phone number on the business card.

1. Introduction

Business cards are still widely used to share contact information in the business world. But they are often lost and the business people don't have time to put all the contact information in their computer. So it would be helpful if one could just take a picture of the business card and the contact information is stored automatically. With this project, we present a first step towards this direction.

An image of a business card is taken as an input, and the program highlights the area on the card where general contact information can be found as well as the e-mail address and the phone number. To do so, the image has to be pre-processed first to find the boundaries of the card. The card is then transformed in order to align it horizontally and it is cropped to simplify further analysis. As a next step, the text areas on the card need to be detected, which are considered as the area of general contact information. Inside of these areas, the individual characters are found and separated. At this stage, each character should have a clear bounding box assigned to it, which is important for the last step: the optical character recognition. In this project only two characters are being recognized: "+" and "@". They indicate where the phone number and the e-mail address can be found on the card and these text areas are highlighted in the output image.

The pre-processing part was done by Sara, so this report mainly focuses on the text detection, the character segmentation and the optical character recognition. The implementation of these tasks is explained in more detail in the following chapters.

2. Text Detection

2.1. Failed Approaches

The first approach to detect the area of text in the image was to look at the histogram. Text is normally dark and occurs often on the card (compared to other dark pixels), so I tried to find a dark pixel value that occurs most often. But if there is a dark pattern or decoration on the card, this region will be much bigger than the text area. So I added a condition that the dark pixel value which should be detected doesn't cover too much of the whole card (since the text area is relatively small compared to the whole card). The problem with this method is that not the whole text has the exact same pixel values, so one needs to consider a range of values. But there are other pixels in the image which also lie in this range, so not only text areas are detected.

The next approach was based on the fact that characters have a lot of vertical lines. In order to detect these vertical lines, a Sobel mask for vertical edge detection was used. This detects all the vertical lines of the text, but also other lines in the image. The vertical lines in text areas have the same distance between each other, so they have a certain frequency. Using a gaussian bandpass filter, only the frequency of the text should be passed [1]. The difficulty is to find the frequency of the text, which is also dependent on the font and the style of the characters. The font and the style of the text can also vary in the same business card, so it is not possible to use only one bandpass filter. Additionally, the passed frequencies are also found in other areas, so this approach was also rejected.

2.2. Final Approach

A characteristic of a line of text is that the characters are horizontally aligned and have a small, constant distance between each other. By dilating the image with a horizontal line as a structuring element, the characters in a text row are

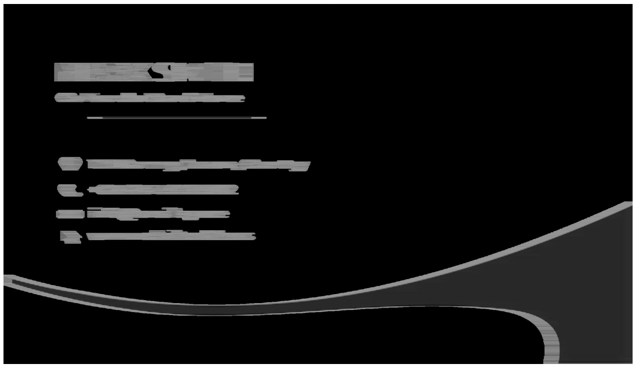

Figure 1. Business card after dilation with horizontal line structuring element. A threshold was applied to the image before dilation so that only dark pixels were kept and bright pixels are set to 0.

connected (see fig. 1). Then, bounding boxes around all connected pixels in the image are found. Due to the dilation also some other features than text are connected, so not all the bounding boxes contain text. If a box is too small or too large compared to the size of the card it is not considered as text. If no characters are found inside the box in the character segmentation step, the box is also not considered as a text box. The resulting boxes are then marked in the image as "contact information" as shown in fig. 2.

This method is based on dilation, which is a common image processing technique [1]. This makes it simple and easy to understand. Furthermore, it reliably finds the text areas if the card is accurately aligned. But there are also some drawbacks. If a pattern has similar characteristics as the ones discussed for text, it will also be detected. For most of the business cards this is not a problem, but sometimes errors could occur due to this issue. Another problem is that if the contrast between the text and the background isn't high enough, the background will be considered as connected to the text. An assumption that needs to be fulfilled is that the background is bright and the text is dark, so if the background is darker than the text this method will not work either. This could be resolved by analyzing the histogram at the beginning. If the majority of the card is dark one could assume that the background is dark and the text is bright. But this wasn't the case for all the considered business cards, so it was not implemented in this project.

3. Character Segmentation

This step takes the bounding boxes and the cropped images of each text fields as an input. The cropped images are converted to binary images. In order to separate the individual characters inside each text field I first used opening to be sure that there will be a gap between characters if they are overlapping. But this also produced holes and gaps inside the characters. To ensure that the characters are not divided

Figure 2. Business card with highlighted areas containing contact information.

into parts, closing instead of opening was then performed. For the separation of the characters, I used the vertical projection method [1]. This method adds up each element of the columns of the text box, which results in a horizontal profile. For our application it is only important if all the values in a column are zero or not. An example of such a profile is given in fig. 3. If the sum of the column is greater than zero, white pixels are displayed and black pixels if the sum is equal to zero. So each change from black to white in the profile indicates the beginning of a character and a change from white to black indicates its end.

With this method a problem arises if the characters touch the boundary of the text box. In this case, the number of starting and ending points don't match and the separation fails. This was handled by introducing starting or ending points at the boundary of the box if necessary.

If the resolution of the image is too bad which causes the characters to touch each other, they are not separated. So if a character after separation is still too wide compared to its height, there are probably more than one characters combined. In this case, the corresponding image is binarized with a threshold that is more restrictive than before and opening is performed to disconnect the characters. After this step, the same vertical projection as before is performed, but only on the affected characters. Still not all the characters can be perfectly separated, especially if bold characters are close together (as the two characters RV in

Figure 3. Vertical projection method. The horizontal profile is shown underneath the characters. Note that the height of the profile is extended for better visibility.

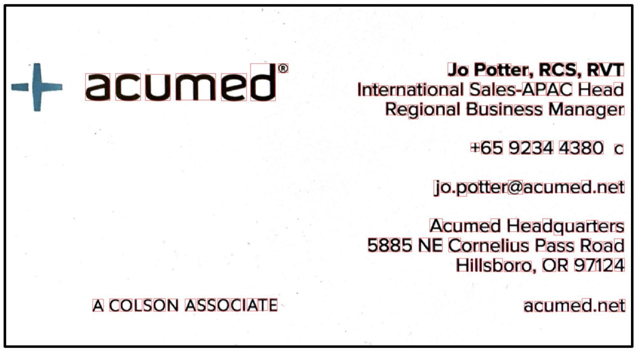

Figure 4. Result of character segmentation.

the upper right corner of fig. 4). This issue could be solved by making the size of the structuring element used for opening adaptive to the density of dark pixels in the box. The more dark pixels there are in the box, the greater the overlap of the characters will be and a bigger structuring element should be used. But the performance is good enough to separate all "+" and "@" for the considered business cards.

Even though the characters were separated, the performance of the character recognition explained in the next section was not good enough to recognize the desired characters. One possible reason for this was that the characters were not exactly bounded in the other direction, the vertical direction. The height of the character bounding boxes was still the same as for the whole text line. So I applied the projection method again, but this time the sum over the rows instead of the columns was taken (horizontal projection). As a result, most of the characters are now bounded by an exact bounding box as seen in fig. 4, which is important for the optical character recognition.

4. Optical Character Recognition

As already stated before, we are only recognizing the characters "@" and "+" in this project. In the following, the details of the recognition of these two characters are explained.

4.1. Recognition of @

The approach to recognize these characters was to erode each character in the image with a structuring element shaped like an @. This structuring element was created using an @ character from one of the business cards and transformed to a bitmap (the mask is shown in fig. 5 c). The mask is always resized to the size of the character that is currently analyzed. If the image of the character is eroded with this mask, it should yield non-zero values in the center of the character image if the character is an @ (see fig. 5 d). But many @'s on the business card couldn't be recognized up to this point, because the mask did not match the

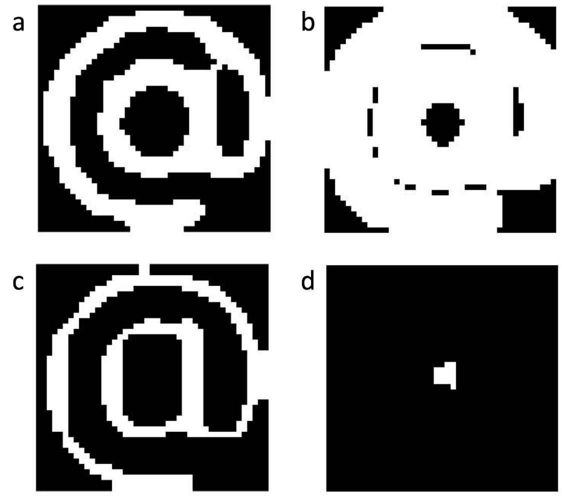

Figure 5. Steps of @-recognition. (a) Cropped and binarized character image as it appears on the business card. (b) Dilated character image of the business card (with a square mask). (c) @-mask used for erosion. (d) Result of erosion of the dilated image with the @-mask.

@'s completely. As an alternative approach, the difference between the character images and the mask was taken and the character with the largest overlap of mask and character was considered to be an @. The problem with this approach was that other characters (especially bold characters) often had more overlap than the @. It also recognizes exactly one @ in each card, independent of the number of @'s that are actually present. The next idea was to refine the erosion explained before. If no @ was recognized on the whole business card in the first iteration, the character images are dilated by a square mask. The size of this mask is dependent on the size of the character. This makes the characters much thicker (see fig. 5 b), which increases the chance that the @ mask fits completely into the characters.

4.2. Recognition of +

For the recognition of the character +, an erosion like the one explained for the @-recognition finds too many matches in other characters as well. For this reason, a Hit-Miss filter was used [1]. To perform this operation, two masks are needed: one mask that should be fully contained in the character (as for the @-recognition), and an additional mask that should not at all be contained in the character (see fig. 6 b and c respectively). These two masks were also generated using a + character from one of the business cards. With the mask that should be contained in the character, an erosion of the foreground is performed. With the other one the background is eroded. Only if both of these erosions yield non-zero values in the center of the character image, a + is recognized.

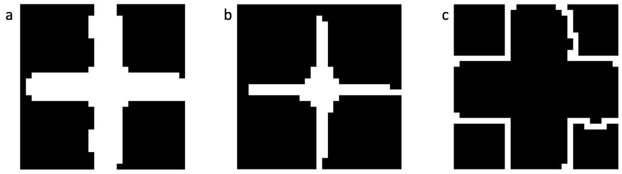

Figure 6. Hit-Miss filter for + recognition. (a) Cropped and binarized character image as it appears on the business card. (b) Mask for foreground erosion. (c) Mask for background erosion.

Similar as for the @-recognition, the hit-miss operation is refined if no + character is found in the first iteration. If so, the mask which is used for erosion of the foreground is eroded by a small structuring element. The other mask is dilated and the previous mask is subtracted from it in order to enlarge the area that is passed and to keep a thin edge of the mask.

5. Results

If an @ is recognized, the text box in which it occurs is considered to be an email address and it is labeled accordingly. If a + is detected, the box is labeled with telephone number and both areas are highlighted in the image of the business card, which can be seen in fig. 7. The presented methods to recognize the characters @ and + are recognizing most of the @ and + in the tested business cards, as shown in table 1.

Tested cards: 13

Character	Amount tested	Recognized	Not recognized	Falsely recognized
+	23	19	4	0
@	12	11	1	0

Table 1. Statistics of results.

6. Shortages

Even though the presented methods show good results, they are prone to noise. For example if there are noisy pixels between two characters, they can't be separated correctly. This would then also lead to failing of the recognition of the characters. Another shortage is that there are some parameters like filter sizes of erosions and dilations, thresholds for converting the images to binary images as well as smoothing coefficients which may need to be adjusted for certain applications. The + and @ masks that are used will not match all the different styles that these characters can occur in, so they could be refined or extended by other masks that can also be used to find potential matches. At the beginning, some assumptions on the size of text in the images as well as the spacing of the text were made. These may be true for most of the business cards, but certainly not for all of them.

Figure 7. Result after the text fields of the recognized @ and + characters are highlighted and labeled with e-mail address and telephone number respectively.

7. Learning Outcomes and Future Work

As I implemented most of the functions used during this project myself, I really learned a lot in traditional image processing. I was motivated to make use of the principles and methods that we have seen during the lectures. I was spending many hours on refining the algorithms to get better results and it was very interesting to think about how to achieve this. To do so, one really needs to understand how the applied methods work and how they can be adjusted to fit the requirements of the projects. Overall, the project was a lot of work, but I enjoyed working on it and by considering the good results that were achieved, the hard work paid off. One could definitely take the project one step further by trying to detect more different characters and put them together to words. Then it would be possible to save the contact information on the business card to the phone or the computer automatically.

References

[1] R. C. Gonzalez and P. Wintz. Digital image processing(book). *Reading, Mass., Addison-Wesley Publishing Co., Inc.(Applied Mathematics and Computation*, (13):451, 1977.