CASE 80

Printed Letter Inspection Technique Using the MTS

Abstract: There are many products, such as computers or mobile telephones, whose operation buttons or keys are printed with letters. The quality of these printed letters is inspected visually to find defects. However, it is difficult to clearly determine the quality or defects because of defect types. This problem has not been solved. It is therefore necessary to develop an automatic inspection method for quantification of quality. In this study the Mahalanobis–Taguchi System (MTS) was used for this purpose. The experimental results agreed well with the results of human inspection.

1. Introduction

Since in recent years the number of cellular phones has increased rapidly, not only are size and weight reduction under active research but also the phone's appearance. More attractive, easy-to-understand printed characters and labels are highly sought after.

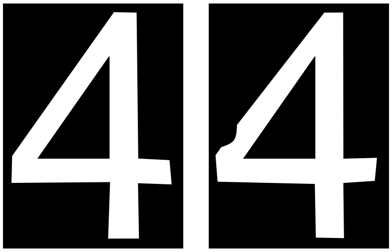
A keypad character is normally integrated into a keypad. While the size of a character changes to some degree with the type of cellular phone, the digits from "0" to "9" and symbols like "*" and "#," which were studied in our research, are approximately 3 mm × 2 mm with a linewidth of 0.4 mm. On the other hand, Kana (Japanese syllabaries) or alphabetical characters are about 1 mm × 1 mm with a linewidth of approximately 0.16 mm. While fade resistance is required for these characters, for characters on keypads, a material and printing method that make a back light pass through are used such that a cellular phone can be used in a dark environment.

The keypad characters are printed in white on the blacklike background. Various defects are caused by a variety of errors in a printing method. Figure 1 shows a typical example. While Figure 1*a* is judged as a proper character, Figure 1*b*, having a chipped area on the left, is judged as a defect. In fact, a part of the line is obviously concave. As long as this defect lies within a range of 10% of a linewidth, it does not stand out much. However, if it exceeds 30%, a keypad is judged as defective because the defect gives a poor impression on a character. In addition to this, there are other types of defects, such as extreme swelling of a line, extremely closed round area of an "8," or additional unnecessary dots in a free space on a keypad.

2. Image-Capturing System

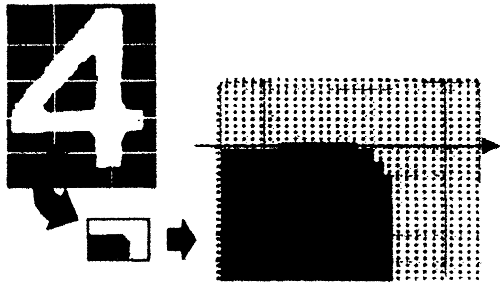
In order to capture a highly accurate image of a character, we organized an image-capturing system with a CCD (charge-coupled device) camera, image input board, and personal computer. Figure 2*a* shows a photo of an image-capturing device, and Figure 2*b* outlines the overall system. The CCD camera used for our research can capture an image with 1000 × 1000 pixels at a time. Using this image-capturing system, we can capture a 3 mm × 2 mm character as an image of 150 × 100 pixels and a 0.4-mm linewidth as that of 20 pixels.

Because the light and shade of a captured image are represented in a 256-level scale, we binarize the image data such that only part of a character can be extracted properly. By setting up an appropriate threshold, we obtained a clear-cut binary image. Furthermore, we performed a noise-reduction



(*a*) (*b*)

Figure 1
Printed character



(*a*) Horizontal Direction (*b*) Vertical Direction

Figure 3
Extraction of feature values

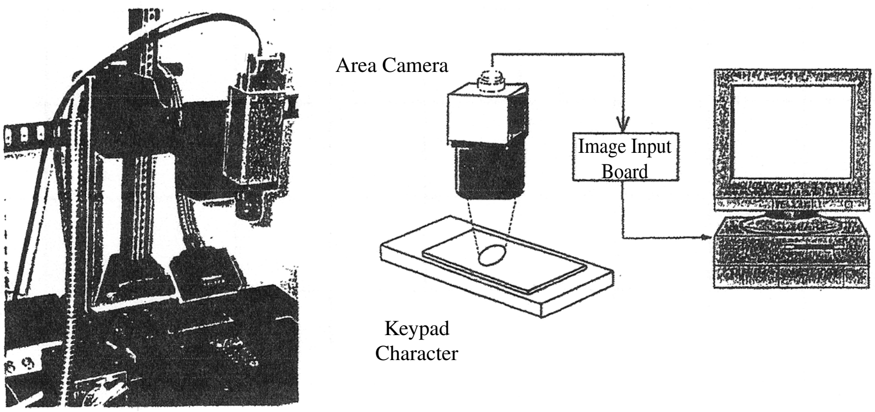
treatment on the image. As a result, an individual character or sign can be picked up, as shown in Figure 3.

3. Base Space

A judgment on whether a certain character is proper or defective is made in accordance with the inspection standard. However, as discussed above, it is quite difficult to prescribe detailed inspection standards for all defect types, such as a chipped area or swelling, or all defect cases on a character. In addition, even if a printed character is too bold or fine on the whole, when it is well balanced, we may judge it as good. The overall shape of a character may be considered as one of the judgment factors. Then the final judgment depends greatly on a human experience.

Therefore, assuming that the inspector's judgment is correct, we gather proper and defective characters. Two inspectors participated in the experiment. The results judged by the inspectors as correct were defined as standard data. The proper characters collected were 200 sets of digits from "0" to "9" and the symbols "#" and "*".

To use the MTS method we extracted some multidimensional feature values that are considered necessary to judge whether a certain character pat-



(*a*) (*b*)

Figure 2
Character inspection system

tern is proper or defective. In our research we used as a feature value differential and integral characteristics which are applied to character recognition.

A character obtained as a binary image is, for example, represented in a pixel image 150 in height × 100 in width. Since the extract of feature values from this original image results in a considerable number of feature values, we divided the image into appropriate-size unit regions. For example, we split them up into many unit regions, each with 30 × 30 pixels. These were divided into 5 × 4 unit regions. Then, for each unit region, we calculated the following feature values.

1. For 30 lines in a horizontal direction, we selected the number of areas that change from white to black, or vice versa, as a differential characteristic. In addition, the sum of lengths of areas occupied with a character data was chosen as an integral characteristic.
2. Similarly, for 30 lines in a vertical direction, we computed differential and integral characteristics.

Thus, the total number of feature values adds up to 120, or the sum of all differential and integral characteristics. This workflow and an example of feature values are shown in Figure 3 and Table 1, respectively.

4. Results of Inspection

After creating software that covers the aforementioned workflow, ranging from image capturing to character extraction, character division into unit regions, extraction of feature values, generation of base space, and calculation of Mahalanobis distances, we performed an experiment. For the division into unit regions, the software automatically split up an image into unit regions so that each consists of 30 × 30 pixels. Therefore, the number of unit regions varies with the size of a character or sign. For instance, when the size of a character is 150 pixels in height × 100 pixels in width, its image is divided into 5 × 4 unit regions, whereas it is split up into 4 × 3 unit regions in the case of a character of 120 pixels in height × 80 pixels.

Figure 4 shows an example of displaying an inspection result. A binarized character is indicated on the left of the screen. On the right, a Mahalanobis distance for each unit region is shown. In addition, if a Mahalanobis distance exceeds a predetermined threshold, the corresponding area in the character on the left lights up. On the other hand, for all Mahalanobis distances surpassing 1000, 1000 is displayed.

As discussed before, using 200 sets of characters that have been preselected as proper by the two in-

Table 1
Feature values extracted from a character

	Differential Characteristic	Integral Characteristic		Differential Characteristic	Integral Characteristic
1	0	30	1	2	15
2	0	30	2	3	16
3	0	30	3	3	15
⋮	⋮	⋮	⋮	⋮	⋮
27	1	15	27	2	27
28	1	14	28	1	26
29	2	14	29	2	27
30	1	12	30	2	27

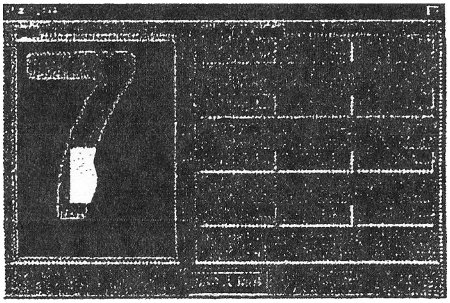
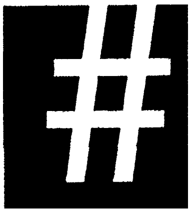


Figure 4
On-screen inspection result for character recognition by MTS

spectors, we created a Mahalanobis space for each unit region of each character or sign. Then, with 100 sets of characters, including both proper and defective, we performed an inspection experiment.

In the example shown in Figure 4, we can see a swelling in the lower part of a digit "7" whose Mahalanobis distance is computed as 10.337. All other regions have a smaller distance. Figure 5 indicates other examples together with a table of Mahalanobis distances. In the figure a small swelling can be found in the bottom left part of the symbol "#." In fact, the corresponding Mahalanobis distance amounts to 10.555. Although one of the right-hand regions shows the value of 3.815, because the threshold is set to 4, the region is judged as proper on the display. For the digit "0," due to dots scattered in the bottom-left and top-right regions, both of the Mahalanobis distances exceed 1000. The reason for displaying these enormous values is that there is no dot in the standard data.

As a next step, in Table 2 we show the relationship between an inspector's judgment and a Mahalanobis distance based on the inspection of 1000 unit regions. Table 2 indicates the correspondence between proper-or-defective judgment by one inspector and the Mahalanobis distance, irrespective of a judgment based on a Mahalanobis distance. Based on this table, we can see that, for example, if we select 4 as the threshold for a proper-or-defective judgment [(a) of Table 2], the error of judging a defect as proper results in 22% = (4 + 14)/81, and the contrary error of judging a proper outcome as defective turns out to be 23% = (22 + 36)/249. When setting the threshold to 3 [(b) of Table 2], the error of judging a defect as proper is reduced to 5%, whereas the opposite error of confusing a proper character with a defect is increased.

	0.009	1.006	0.582	0.000
	1.471	0.914	0.984	0.237
	0.784	1.189	0.205	3.815
	10.555	0.668	0.488	0.117
	0.200	0.398	0.223	0.000

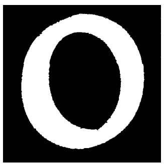
	0.789	0.523	1000.000
	0.719	0.491	0.533
	5.318	0.958	0.336
	1000.000	0.411	0.183

Figure 5
Inspection results

Table 2
Comparison of inspector's judgment and Mahalanobis distance

	Inspector's Judgment		
Mahalanobis Distance	**Proper**	**Defective**	
$0 \leq MD < 1$	16	0	
$1 \leq MD < 2$	78	4	
$2 \leq MD < 3$	62	0	← (a)
$3 \leq MD < 4$	35	14	← (b)
$4 \leq MD < 10$	22	19	
$10 \leq MD$	36	44	
Total	249	81	

5. Conclusions

As shown in Table 2, since the Mahalanobis distances indicate a clear-cut trend for a judgment on whether a certain character is proper or defective, the evaluation method grounded on the MTS method in our research can be regarded as valid to detect a defective character. Despite the fact that this method needs to be improved further because it sometimes judges a proper character as a defective, we have successfully demonstrated that our conventional evaluation of quality in character printing, which has long relied heavily on human senses, can be quantified.

The following are possible reasons that a proper character is judged as defective:

1. Since the resolution of character image capturing (a character of 3 mm × 2 mm is captured by an image of 150 × 100 pixels) is regarded as insufficient, higher resolution is needed. Although 200 sets of proper characters were used in this research as the data for creating a Mahalanobis space, the number is still insufficient to take in patterns that are tolerable as a proper character. At this time, we believe that the second problem is more significant. Considering that the research on character recognition deals with 3000 handwriting characters for a standard space, we need to collect at least 1000 characters as the standard data.
2. The inspection processing time in our study, which ranges from extraction of feature values to calculation of a Mahalanobis distance, has resulted in less than 0.1 second. In addition, preprocessing time for data processing and binarization of a captured image is needed. However, since our current visual inspection requires 10 to 15 seconds for one sheet of keypads, we can expect to allocate a sufficient amount of preprocessing time.

In this research we have not sieved necessary essential feature values. On reason is that because a character is divided into unit regions, it is quite difficult to analyze comprehensively the priority of each of 120 differential and integral characteristics defined in each unit region, and the other is that 120 feature values do not lead to an enormous amount of calculation.

Reference

Shoichi Teshima, Dan Jin, Tomonori Bando, Hisashi Toyoshima, and Hiroshi Kubo, 2000. Development of printed letter inspection technique using Mahalanobis–Taguchi system. *Quality Engineering*, Vol. 8, No. 2, pp. 68–74.

This case study is contributed by Shoichi Teshima, Dan Jin, Tomonori Bando, Hisashi Toyoshima, and Hiroshi Kubo.