Fault Detection of Label on a Curved Surface by Image Template Matching with Hierarchical Affine Transformation

Shinobu Kakiuchi, ○Yoshiki Sato,Hiroshi Maeda, and Norikazu Ikoma
Faculty of Engineering, Kyushu Institute of Technology
1-1 Sensui-cho, Tobata, Kitakyushu, 804-8550, Japan
email:hmaeda@comp.kyutech.ac.jp

Abstract – It, at present, is difficult for machine to automatically detect small faults of a label pasted on a curved surface such as a PET bottle. Thus, many hands do that in a factory. In this paper, we present a new, simple fault detection method for a PET bottle label by image processing. It contains hierarchical affine transformation that is formulated as a combinatorial optimization problem. Three combinatorial optimization methods are applied to this problem. The effectiveness of the methods is experimentally confirmed for 6 test images.

Ⅰ. INTRODUCTION

At present, it is still difficult for machine to automatically detect small faults on a label pasted on a curved surface such as a PET bottle. Thus, many hands do that in a factory. These works are overload for human eyes and then it is related to miss-detection of fault. Therefore, automatic fault detection is strongly requested.

Now, this paper presents a detection method of small faults on a label pasted on a PET bottle by a simple image processing based on template matching. We propose a new template matching method with two-dimensional affine transformation for the divided subparts of a template image. A real label image might be transformed 3- dimensionally in comparison with a template image. Usually affine transformation is used to absorb this transformation but it is not easy to determine the parameters of 3-dimensional affine transformation because of the number of parameters. Furthermore, we have to considerer the consuming time for detecting fault from the viewpoint of a real time test. Then, considering that the transformation is little to the direction of depth, it is possible to apply 2-dimensional affine transformation to the divided subparts of a template image, which might be hierarchically divided into further small areas. The identification of the affine transformation parameters is formulated as a combinatorial optimization problem. The three optimization methods of complex method[1], tabu search[2], and genetic algorithm[3] are applied and then evaluated from the viewpoint of computation efficiency.

The proposed fault detection method is experimentally examined for the 6 labels with artificial faults such as partial distortion, adhesion of particles, and adhesion of hair.

Ⅱ. FAULT DETECTION METHOD OF LABEL

A. Outline of fault detection process

Test subject, here, is a label pasted on a PET bottle. The label surface of a PET bottle, which is almost parallel to camera surface, is taken a digital image by a line camera shown in *Fig.* 1. This image is roughly processed as the followings:

1) Extracting label part from the image.
2) Dividing the label part into several subparts.
3) Matching a template image to a test label in each subpart.
4) Judging fault as the first stage.
5) Dividing the subpart into further small areas in case where the first judgment is not clear.
6) Matching again a template image to a test label in each small area.
7) Judging finally the fault.

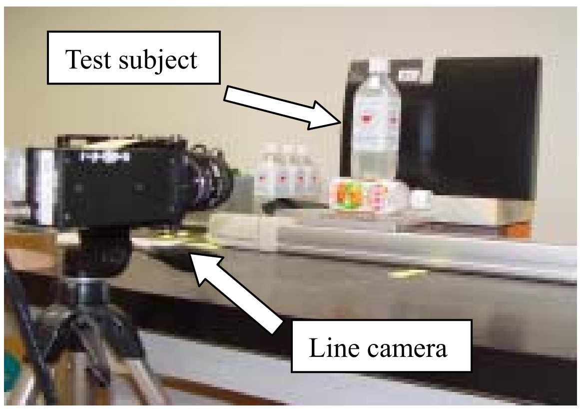


Fig. 1 Line camera system

B. Extraction of label part and division into subparts

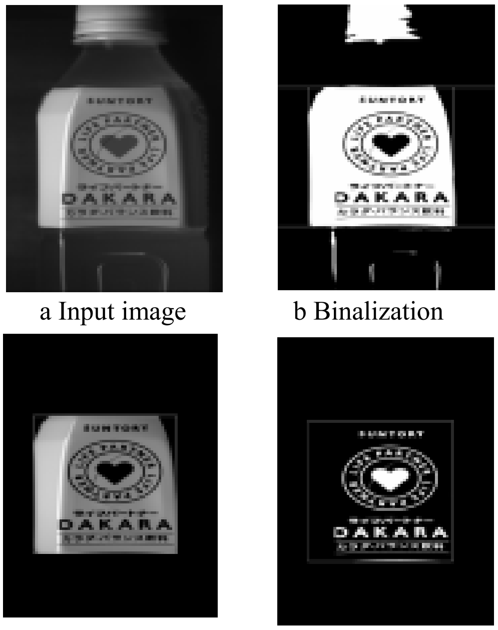
A label surface of a PET bottle is taken a digital image as shown in *Fig.* 2a. The following 5steps are done in order to extract the label part from the image.

Step 1: Binalization to separate a label part and a background part from the image by using *P*-tile method on brightness distribution (*Fig*. 2b).

Step 2: Extraction of a label part by using *P*-tile method based on brightness distribution on a horizontal axis and a vertical axis (*Fig*. 2c).

Step 3: Extraction of characteristics such as letter, symbol, figure etc. in a label (*Fig*. 2d).

Step 4: Division of a label part into three subparts (*Fig.* 3) by a threshold in brightness distribution with Gaussian filter of the binary image shown in *Fig.* 2d.



a Input image　　b Binalization

c Extraction of label　　d Characteristics

Fig. 2 Preliminary image processing

Fig. 3 Three sub parts

C. Template matching

First, a template image for each subpart is prepared from a normal label. The precise register between a template image and a test one is essential in order to correctly detect small faults by a template matching. The test image might be 3-dimensionally transformed, which is quite little to the direction of depth, in comparison with the template image. Then we apply 2-dimensional affine transformation to the divided subparts of a template image. When a rotation angle is small, the 2-dimensional affine transformation is approximated as

$$\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{bmatrix} \alpha_x & \theta \\ -\theta & \alpha_y \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{1}$$

where (X, Y) is a transformed coordinate and (x, y) is an original coordinate, α_x and α_y are scale parameters, and d_x and d_y are parallel transformation parameters. In a digital image, (X, Y) and (x, y) take integer values, so that the parameters of the affine transformation must also take integer values. Then, the parameters $\boldsymbol{P}=(\alpha_x, \alpha_y, \theta, d_x, d_y)$ is transformed to the integer parameters $\boldsymbol{m}=(m_1, m_2, m_3, m_4, m_5)$; m_1 is the number of pixels moving to a horizontal axis by a scale transformation, m_2 is the number of pixels moving to a vertical axis by a scale transformation, m_3 is the number of pixels moving by a rotation, m_4 is the number of pixels moving to a horizontal axis by a parallel movement, and m_5 is the number of pixels moving to a vertical axis by a parallel movement.

Let $R(i, j)$, $I(i, j)$, and $T(\boldsymbol{m})$ be a template binalized image, an input test binalized image, and 2-dimensional digital affine transformation, respectively. Then, the parameter identification is represented as the following combinatorial optimization problem:

$$\text{Minimize} \quad f(\boldsymbol{m}) = \sum_i \sum_j \left| T(\boldsymbol{m}) R(i,j) - I(i,j) \right|^2, \tag{2}$$

$$\text{subject to } m_L^k \le m_k \le m_U^k,\ m_k \in Integerset,\ k = 1,,,5.$$

where m_L^k and m_U^k are given and mean the minimum and the maximum values. Here, three combinatorial optimization methods, complex method, tabu search, and genetic algorithm are applied to this problem.

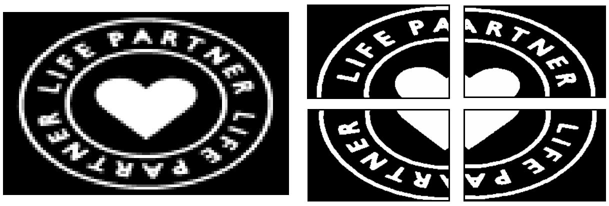
D. Judgment of fault

After the optimum parameters are identified, we can obtain a difference image between an affine transformed template binalized image and an input test binalized image. The difference image is processed with dilation-erosion in order to absorb quantumization errors around edges and then the averaged error, that is the averaged number of white pixels, is calculated. Let it put D_1. By using D_1, the first stage judgment J_1 is given as

$$J_1 = \begin{cases} \text{OK}, & D_1 \le D_1^* , \\ \text{NG}, & D_1 > D_1^* \end{cases} \tag{3}$$

where D_1^* shows a threshold for judgment.

In case of J_1 = NG, there are two possibilities. One is that the subpart of a label is fault. Another is that the subpart is not fault but D_1 exceeds a threshold. The latter is the case that the transformation (1) is not sufficient to absorb 3-dimensional deformity of a label. Then, we introduce further segmentation of a subpart, which divides a subpart into 4 areas at the center of gravity as shown in *Fig*. 4. Let the *i*-th area put A_i. The combinatorial optimization problem (2) is resolved again for A_i (i=1,,,4) and then the optimum parameters $\boldsymbol{m}_i$ (i=1,,,4) are obtained.



a. Subpart b. Divided 4 areas

Fig. 4 Segmentation of a subpart

Now, we get another difference image by aggregating the difference images of the 4 areas. Let its averaged error put D_2. By using D_2, the second stage judgment J_2 is given as

$$J_2 = \begin{cases} \mathrm{OK}, & D_2 \le D_2^*, \\ \mathrm{NG}, & D_2 > D_2^* \end{cases} \qquad (4)$$

where D_2* shows a threshold for judgment. By J_2, we could almost judge the fault including 3-dimensional deformity of a label. However in case of J_2 = NG, it might not be distinguish whether D_2 dues to fault or to other factors, e.g., quantumization error and remained distortion.

Then, we furthermore define error change rate F as

$$F = \frac{|D_1 - D_2|}{D_1}, \qquad (5)$$

and introduce the third stage judgment J_3 as

$$J_3 = \begin{cases} \mathrm{OK}, & F \ge F^*, \\ \mathrm{NG}, & F < F^* \end{cases} \qquad (6)$$

where F* shows a threshold for judgment. Note that a sign of inequality in equation (6) is different with (3), (4).When $J_1 = J_2$ = NG by the factors without fault, F taking a big value is expected. This is because D_2 becomes pretty small comparing with D_1 by the further segmentation. Thus, it is highly possible to judge OK by J_3.

III. EXPERIMENT

Test labels, which are the same as template (A), with hair (B), with particle (C), with particle (D), with partial distortion (E), and normal (F) are shown in *Fig*. 5. Here, fault is in a part wish a circle. These images have 1500×2000 pixels with 256 grey scales. The locations of faults, which are represented as NG, are summarized in Table 1.

Fig. 6, for instance, shows the three subparts of image (B) extracted from the whole label are shown in *Fig*. 6.

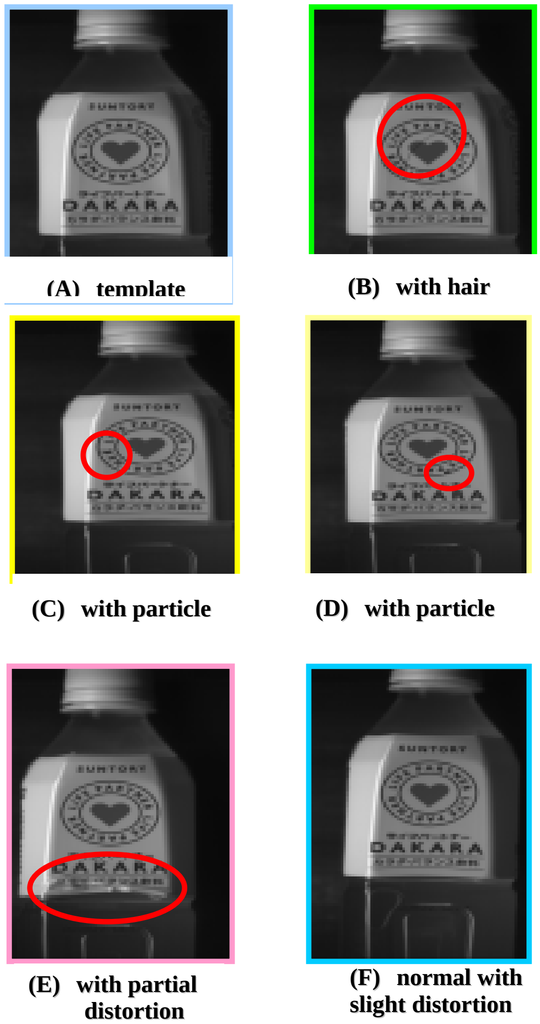


Fig. 5 6 test images

Table 1 Location of fault

	S_1	S_2	S_3		S_1	S_2	S_3
A				D		NG	
B		NG		E			NG
C		NG		F			

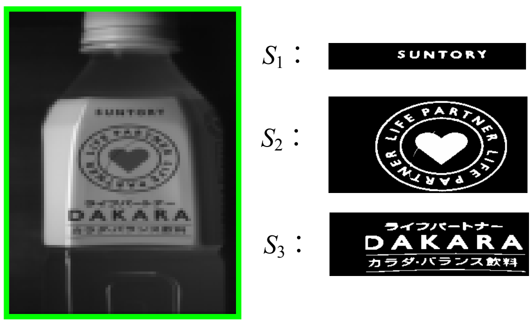


Fig.6 Extraction of 3 subparts of image (B)

B. Results of combinatorial optimization and of J_1 , J_2

The three optimization methods of complex method (CP), tabu search (TS), and genetic algorithm (GA) are applied. The result of CP with respect to D_1, D_2 is shown in Table 2. Here, the cells with shade indicate NG and underlines indicate correct judgment.

Table 2 Result of D_1, D_2 (CP)

Image	D_1			D_2		
	S_1	S_2	S_3	S_1	S_2	S_3
A	0.00000	0.00000	0.00000	-	-	-
B	0.00000	0.00079	0.00000	-	0.00081	-
C	0.00000	0.00071	0.00000	-	0.00071	-
D	0.00000	0.00221	0.00038	-	0.00151	0.00012
E	0.00000	0.00409	0.03397	-	0.00064	0.02922
F	0.00000	0.01911	0.00439	-	0.00270	0.00044

:*NG* $D_1^* = D_2^* = 0.00$

The difference between an affine transformed template image and a test image without in each subpart is shown in *Fig.* 7a and the difference aggregating further divided 4 areas is shown in *Fig.* 7b. The difference must be zero ideally. It is seen that the difference aggregating 4 areas is pretty less than the one of each subpart. Thereby, dividing further subpart, that is hierarchical segmentation, is very effective. The judgment by J_1 is incorrect with respect to S_2 in images E, F and S_3 in images D, F. The judgment by J_2 is incorrect with respect to S_2 in images E, F and S_3 in image F. Namely, S_3 in image D is correctly judged by J_2. Since the results of TS and GA are almost the same as CP, those are omitted. Table 3 shows computation time for solving the combinatorial optimization. It is seen the computation efficiency of CP is the highest. It seems to be advantageous for CP that the transformation between template image and test image is not so big.

C. Results of the third judgment

The result of the third judgment J_3 using error change rate F is shown in Table 4. S_2 in images E, F and S_3 in images F are further correctly judged by J_3. Using J_1, J_2, and J_3, correct judgment is obtained with respect to fault detection of all the test images. J_3 is quit effective.

Table 3 Computation time of 3 methods (sec)

	Image	CP	TS	GA
S_1	A	11.9	28.9	301.3
	B	10.2	29.8	961.1
	C	11.2	29.4	963.3
	D	8.1	29.3	960.3
	E	7.7	27.4	956.6
	F	4.7	29.2	971.2
S_2	A	95.3	126.6	3889.1
	B	93.7	117.0	4041.0
	C	75.0	148.4	4142.5
	D	82.5	140.3	4039.5
	E	79.2	125.8	4552.6
	F	104.0	133.8	4146.7
S_3	A	78.6	76.7	1197.4
	B	74.1	85.1	2624.7
	C	89.4	136.8	2705.7
	D	79.2	114.0	2655.2
	E	82.6	79.8	2830.1
	F	91.5	159.8	2691.9
Average of S_1		9.0	29.0	802.3
Average of S_2		88.3	132.0	4135.2
Average of S_3		82.6	108.7	2450.8

Table 4 Result with error change rate (CP)

Image	$\lvert E_1 - E_2 \rvert$			F		
	S_1	S_2	S_3	S_1	S_2	S_3
A	-	-	-	-	-	-
B	-	0.00002	-	-	0.031	-
C	-	0.00000	-	-	0.000	-
D	-	0.00069	-	-	0.315	-
E	-	0.00345	0.00475	-	0.843	0.140
F	-	0.01641	0.00394	-	0.859	0.899

: *NG* $F^*=0.6$

Ⅳ. DISCUSSION ON SPEEDING UP OF COMPUTATION

In the previous section, it was confirmed that CP has an advantage for computation efficiency. Here, we want to discuss the speeding up of CP. Let the number of parameters of an objective function be n. CP algorithm is written as follows:

Step 1: Set up $2n$ points of initial value of parameters.

Step 2: Obtain the most superior point and the most inferior point of an objective function from the $2n$ points. If the difference between the objective function values of the most superior point and the most inferior point is less than a small value, stop the search.

Step 3: Obtain the center of gravity among ($2n$-1) points except for the most inferior point.

Step 4: Search a better point than the most inferior point on a line connecting the most inferior point and the center of gravity.

Step 5: If the better point in ***Step 4*** is found out, replace the inferior point with the better point and go to ***Step 2***. If the better point is not found out, search a better point on a line connecting the most inferior point and the most superior point again, replace the inferior point with the better point, and go to ***Step 2***.

The original CP updates only one point by a search. Then we try to update multiple points by a search. Let the number of updating points simultaneously be k. The results of experiment on k=1 to 4, for S_2 of image D, are shown in *Fig.* 8. In *Fig.* 8, 3 ratio profiles set up k=1 as a standard. The followings are observed:

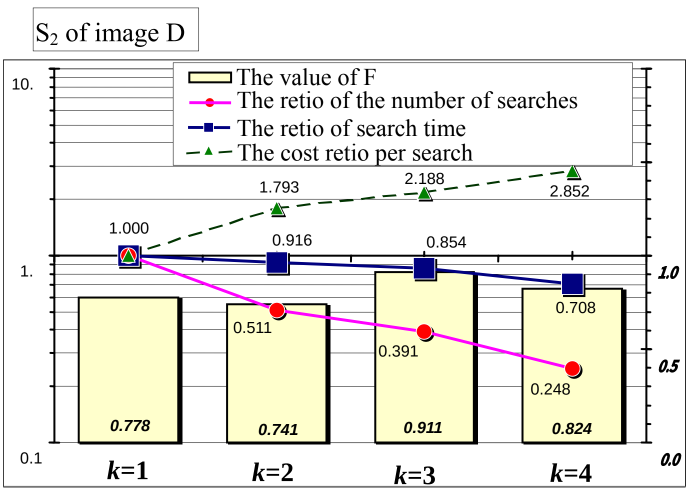


Fig. 8 Result of revised CP

1) The number of search times decreases in inverse proportion to the number of k.
2) The total computation time decreases little by little according to the number of k.
3) The search cost per search naturally increases in proportion to the number of k.
4) The value of error change rate F in k=1 fairly differs from in k=3, 4. It indicates the precision of estimated values of parameters.

Totally, updating two points per search simultaneously seems to be effective. The decrease rate of computation time is about 9% in a sequential processing. The searches for two points are independent each other, so that it is possible to process the searches in parallel and about 50% decrease of computation time may be expected.

Ⅴ. CONCLUSION

In this paper, we proposed a fault detection method for a PET bottle label by image processing containing template matching with hierarchical affine transformation, which was formulated as a combinatorial optimization problem. That well worked for 6 test images with artificial faults. Hereafter, we further experiment for many fault images. On the computational efficiency of combinatorial optimization method, complex method (CP) was quite superior to tabu search and genetic algorithm. Furthermore, speeding up of complex method was somewhat achieved. It, however, is not sufficient yet to test at real time. Making hardware of complex method will be a future issue.

ACKNOWLEDGEMENT

We express our gratitude to Micro Engineering ltd for providing a line camera system.

REFERENCES

[1]Cornelius T. *Optimization techniques*. Academic Press, San Diego, 1998.

[2]Weixiong Zhang. State-Spase search. Algorithms, complexity, extensions, and applications, Spriner, New York. 1999.

[3]D.T Pham and D.Karaboga. Intelligent optimization techniques. London. Tokyo,Springer. 2000.