Template Matching Algorithm For Noise Detection in Cargo Container

Doni Setio Pambudi
Department of Informatics
Universitas Internasional Semen Indonesia
Gresik, Indonesia
doni.pambudi@uisi.ac.id

Ruktin Handayani
Department of Informatics
Universitas Internasional Semen Indonesia
Gresik, Indonesia
ruktin.handayani@uisi.ac.id

Lailatul Hidayah
Department of Informatics
Universitas Internasional Semen Indonesia
Gresik, Indonesia
lailatul.hidayah@uisi.ac.id

***Abstract*— A seaport terminal providing services for inbound and outbound flow of cargo container often has issues with the handling of container. In some cases, the port is sued for damaged container which in fact has not been taken care properly before arriving in the port. Hence an automatic detection system for damaged container is needed. In this research, an algorithm to identify objects in a side-view of a cargo container is proposed. The objects include but not limited to company name of the cargo container, logo, identification code, signs, labels, and damages. By utilizing template matching algorithm, an algorithm to identify objects in container images has been developed. The achieved visual result was satisfactory as well as the computational aspect.**

Keywords—Image Processing, Ship to Shore Crane, camera, container, damage, Template Matching, object detection.

I. Introduction

Being the second most extensive port in Indonesia, Lamong Bay terminal Surabaya serves a strategic role to develop the economy especially for industrial communities in East Java. The port is also a central distribution channel to the entire eastern region of the country. As a world-class terminal, Lamong Bay Terminal always prioritizes efficient and effective service for smooth operation by using the latest and environmentally friendly technology.

One of the problems that occurred and greatly affect the image of service in Lamong Bay Terminal is the handling of the container up to the customer's hand. Container damage happen very often so that many complaints are received by Lamong Bay Terminal. This complaint will subsequently be converted with the nominal loss to be paid by Lamong Bay Terminal to Customer. The losses incurred by damaged container losses can reach up to 2 (two) billion a year. Not only money loss, Lamong Bay Terminal also has to bear poor image that is not good to the quality of service and this will of course affect customer trust.

Damage to the containers that occurred until now is not known for sure whether the it is caused by faults of a series of container removal activities in Lamong Bay Terminal, or the trouble already occurred in the previous port. Field Officers (Tally) have been provided to manually record under the STS (Ship to Shore Crane), but this is not sufficient. Complaints related to container damage are still frequently occurring.

A system to automatically detect damaged container was proposed . The proposed system is called Container Damage Inspection (CDI). But because this tool is very expensive, so up until now there has been no solution to solve the problem of damaged container detection.

In this study, the "Template Matching" method is proposed to automatically detect damage to containers in the lower area of the STS to improve the performance of Tally officers. With this detection, it is expected to know the condition of existing containers arriving at Lamong Bay Terminal before being touched by STS.

Template Matching Algorithm is an algorithm that processes existing image input (container to be lifted) with ideal image (normal container) and image which become output parameter (damaged container). These three images will be compared so that a decision will be made whether the existing container (input image) is declared damaged or not.

II. LITERATURE STUDY

Image processing has been used widely in various applications. Damage detection, character recognition, disease detection, heart beat counting are just a few example from many other useful applications. There are numerous algorithms developed for these purposes. One crucial and common part in these algorithms is noise detection and noise removal.

A. Recent advancement in template matching applications

Template matching is a scientific technique for locating a reference image or any other object inside a larger image [1]. Brunelli and Poggio compared results of face recognition

This research was funded by Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) UISI

between system built based on template matching and another one based on calculation of several geometrical features. It reports that system with template matching performed better [2]. Another application for face detection is presented by Tripathi et.al. where they utilized combined skin color detector in addition to the template matching method [3]. Further more, in face image processing, digital capture of a driver's image is computed with template matching to determine whether there is any signs of fatigue. This is built by Horng et.al. for safety in driving [4]. Lee et.al. developed a template-matching based system to automatically detect pulmonary nodules in images of helical CT scan results [5].

Original template matching algorithm utilizes cross-correlation algorithm. The method is straightforward but slow. Many modifications have been devised to improve its performance. Choi and Kim proposed a method to accommodate the algorithm for rotation since the original one can't handle when the image is rotated [6]. Vanderbrug and Rosenfeld used only subtemplate to reduce the method's computational cost. Only when the subtemplate's matching score exceeds a threshold, the rest of the template is used [7]. Several techniques to reduce the method's drawbacks were compared in [8]. Their approaches includes matched reconstruction residuals, principal component projections, spatial filters, and synthetic discrimination functions.

B. *Recent methods in noise removal algorithm*

Kim et.al. utilized a noise removal techniques based on fuzzy system to detect and remove noise from container digital image in their work for recognition system of shipping container identifier [9].

C. *Recent applications in damaged container detection*

Specific works around seaport container's damage detection include a commercially offered system called Lase Container Damage Inspection (LaseCDI) [10]. LaseCDI provides a detection solution by means of laser combined with Optical Character Recognition(OCR), and high resolution imaging system. This system is claimed to be able to detect various kinds of damages in container such as bulges, dents, tears, holes, etc.

Another commercial product in the market is a system from Visy named Visy Automatic Container Damage Detection System (ADDS) [11]. Similar to LaseCDI, this system combines laser technology, OCR, and visual imaging application.

III. RESEARCH METHOD

Our input are images of container. Using Template Matching Algorithm, the images will be compared with a template, in this case is image of a non-damage container. Fig. 3 depicts the work flow of generating a template.

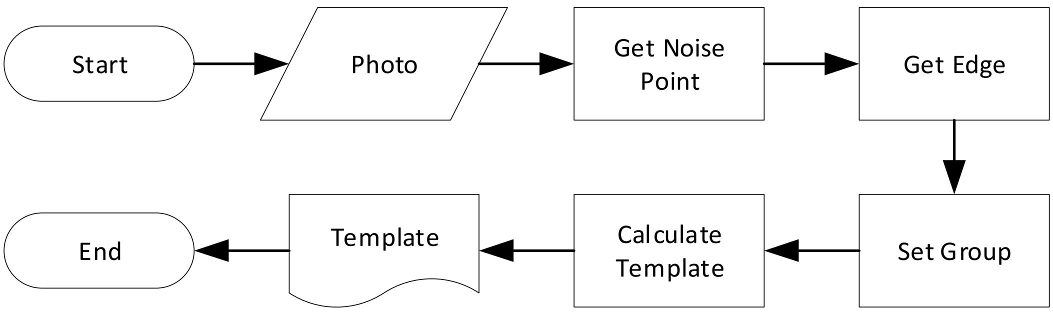

Fig. 1 Flow Chart of Generating A Template

Noise Removal

Container image will have some labels like brands and codes. These objects are considered as noise that will hinder the next process hence need to be removed.

The algorithm in Fig. 1 requires input from container which is taken from perpendicular side. The image will be then divided into grids and search will be performed to find candidate spots for noise. This search is done by computing average image color compared to average color values of the grid. Noise which is considered similar will be grouped into one even though it is from a different grid. This group will be labeled the same. A noise group resulted from the computation will be further computed to produce a data template that can be compared.

Images Requirement

The ideal container image that is required is a photograph that shows the entire container and focuses exactly in the center of the container as illustrated by Fig. 4. The required resolution is at least 1200x500 pixels. In this study, only one side of the container is tested, either the right or left side of the container. Example can be seen in Fig. 5. This study only consider container whose shape is a bar or beam. As for the photographs that are used for testing, we obtained them from Lamong Bay Terminal.

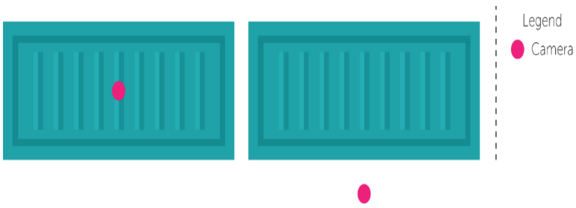

Fig. 2 The position of camera in facing the container

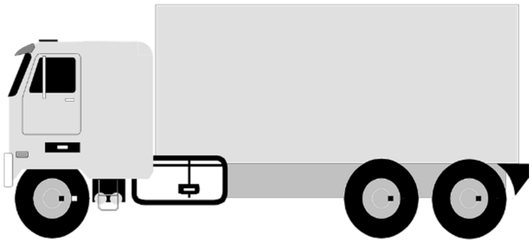

Fig. 3 Side view of the container

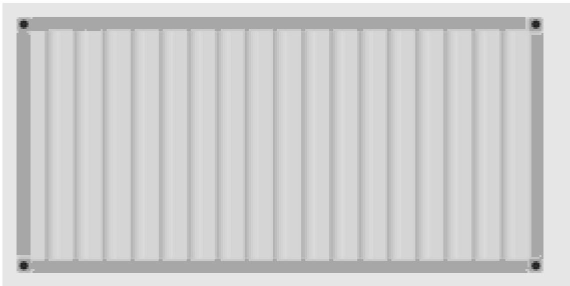

Fig. 4 A side view of container model

Get Noise Point

Each container has a certain base color and generally the existing noise has the opposite color of the base color, so the noise is clearly visible. Fig. 6 shows a clean container side model.

Fig. 5 Flow Chart of Obtaining Container Candidate Noise Point

The designed algorithm calculates the base color of the container by flattening the color from one side. This process is done by summing all components R, G, B and then divide it by the pixels quantity in the image. This is in accordance with (1). The base color which is called α from now onward is a unity of color consisting of components R, G, and B with the value between 0 – 255. This is in accordance with (2).

$$R = \frac{\sum R}{w * h}$$
$$G = \frac{\sum G}{w * h} \quad (1)$$
$$B = \frac{\sum B}{w * h}$$

$$\text{Average Color } (\alpha) = (R, G, B) \quad (2)$$

Data are divided into certain grids in the noise detection process as shown in Fig. 6. Generally a photograph of the container is divided into the same number of grids hence the grid size can be different from each other. The division of this grid is to speed up the process of noise detection. Grid(s) that has the number of noise below the threshold is

not processed since amount that is too low will be considered as noise. In (3) if the average pixel color in the grid (hereinafter referred to as γ) compared with α exceeds the specified threshold then the pixel is considered as noise and added to the amount of noise in the grid.

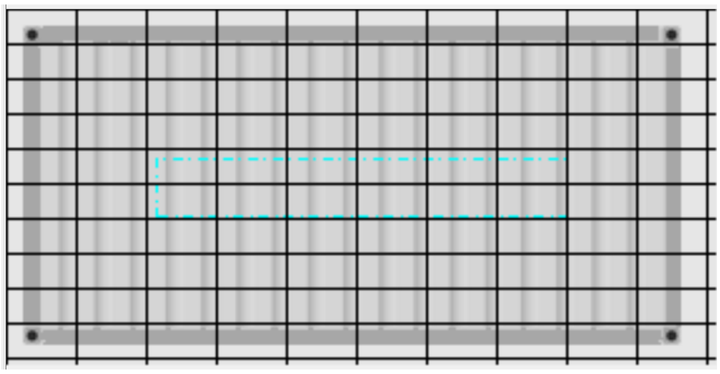

Fig. 6 Grid Built on Container Image

$$\text{Contain noise} = \text{abs}(\gamma - \alpha) < \text{threshold} \qquad (3)$$

In Pseudocode 1 each grid checks each pixel using (3) with the value γ replaced with the pixel value, if the pixel contains noise then it is recorded into the noise list .

```
average ← α
for y = 1 to grid height
   for x = 1 to grid width
      current ← pixel (x, y)
         if abs(current.R – average.R) <= threshold and
            abs(current.G – average.G) <= threshold and
            abs(current.B – average.B) <= threshold
            add to list
```

Pseudocode 1 – Noise Scanning in Grid

Get Edge

Pseudocode 1 produces a list of container noise points, from which the results need to be calculated to determine the edges. This step can be replaced by using edge detection (eg canny, sobel, difference, etc) method. This algorithm uses a simpler way of detecting whether one of its neighbors (left, right, up, down, see Fig. 7) is empty (Pseudocode 2). If it is empty then it means the point is on the edge. Edge determination serves to determine the border of the noise and also to calculate the groups from the container noise points.

	(x, y-1)	
(x-1, y)	(x, y)	(x+1, y)
	(x, y+1)	

Fig. 7 Points coordinate in the image

```
foreach point in list of noise
   get all neighbour from current point
      if one of neighbour is empty
         current.edge ← true
```

Pseudocode 2 - Get edge

Set Group

From the edge point that has been searched in the previous stage , labelling is run according to the relationship between points. If the point is still connected to point(s) in different grid then it is still considered as one group. The categorization of this noise is to unify the noise so that it becomes an object that later can be recognized as one shape.

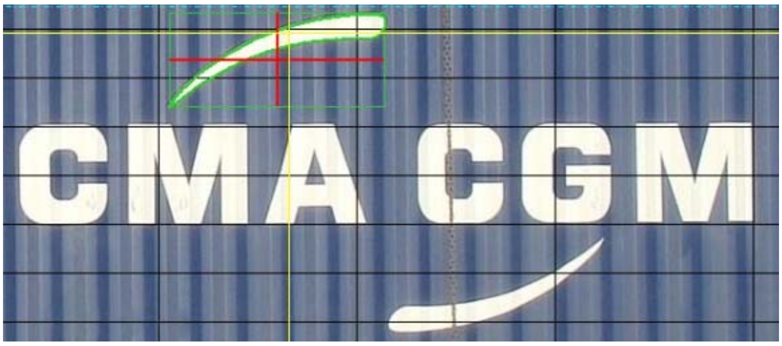

Fig. 8 Noise Object Grouping

The boundary of the noise group area is obtained by finding the lower right corner and the upper left corner using (4) and (5), where x is the value of x for all x coordinates in the same group, and y is the y coordinate value. In Fig. 8 it can be seen in the logo there is a green line. This line indicates the border of a noise group, while the red line is the vertical and horizontal center point of the noise group.

$$b_1 = (\min(x), \min(y)) \tag{4}$$

$$b_2 = (\max(x), max(y)) \tag{5}$$

The step by step procedure of the group setting algorithm is depicted in Fig. 9, where all the points in the image are edge. In Pseudocode 3 the grouping algorithm works by obtaining a point from the edge list obtained from the previous step. This point(s) is then pushed into the stack. This is done until the stack is empty. Neighbors from the popped point is pushed into the stack and set the group label. If it has never been processed, the neighboring point is also set to the group value corresponding to the group from the popped point. The step is repeated until all the points in the edge list are processed. Groups whose number of points is less than threshold are removed from the group list because the group is considered as noise (not the actual container noise).

Calculate Template

Each group of edges will be computed into a template, so if a container has 20 groups of noise it will have 20 noise templates. A template of a container noise contains:

a. Name

Contains the name of the saved template

b. List of information per angle

Contain:

i. Angle

At what angle this information (contains 0 - 359 degrees).

ii. Total point

Contains the number of edge points in this corner.

iii. Total distance

Contains the total distance of the edge point to the midpoint.

iv. List of edge points

List of edge points that have angles equal to Angle

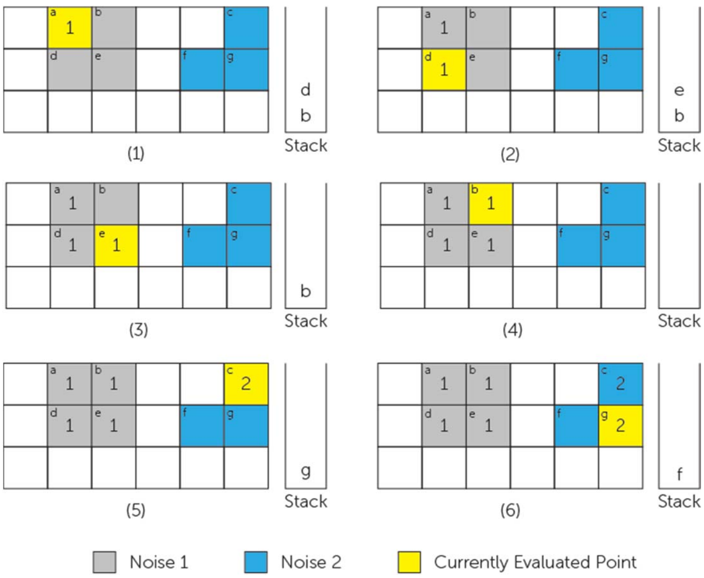

Fig. 9 Steps of Set Group Algorithm

```
create new group
take points in the edge which has not been processed with group from step 1
set all neighboring points to be in the same group
if the number of points less than threshold
        remove group
if not
        add to group list
repeat step 2 for all points
```

Pseudocode 3 – Setting Group of Noise Points

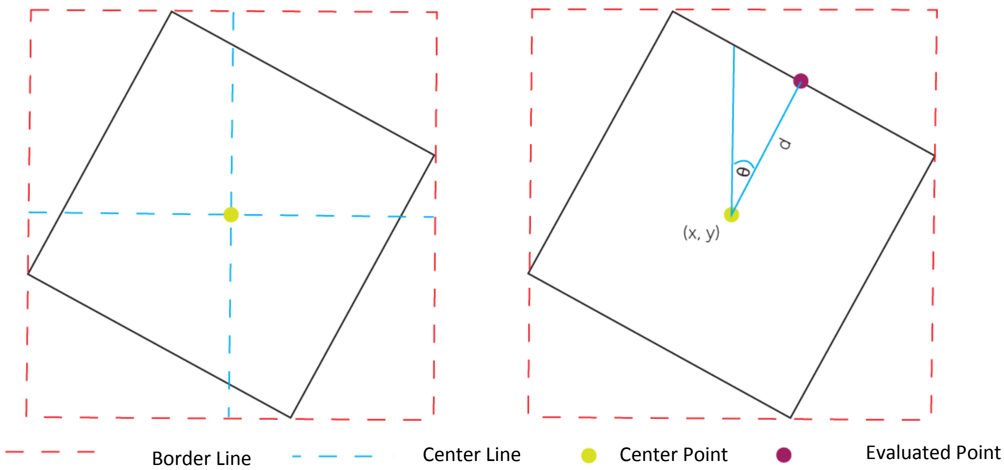

Fig. 10 Noise Template

The interval from an angle within the template has been determined, for instance a 1 degree interval means the number of angle lists is 360. Fig. 10 shows an evaluation point whose distance with the center and angle to the center are calculated. Fig. 11 illustrates in detail an angle in the template where all points that are straight with that angle are recorded (dotted red line). For now the matching process only compares each point angle and total distance from all points. If the difference in the number of points and the total distance is less than the threshold then it is considered the same angle, and if the similarity of all angles exceeds the specified threshold it is considered the same object.

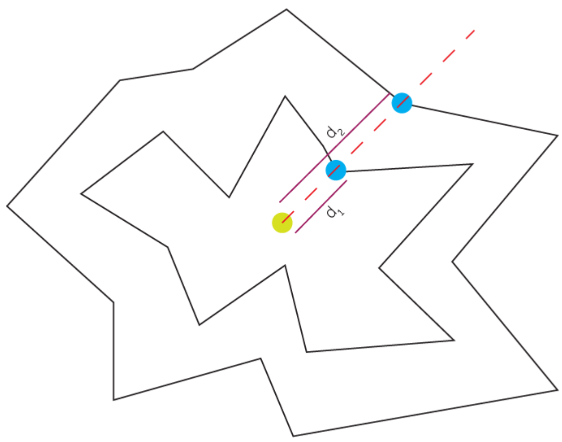

Fig. 11 Data Example in One Angle

V. RESULT AND DISCUSSION

Images Requirement

In order for the algorithm to perform at its best, it requires input image around 1200 pixels minimum wide. The container in the image should also in the side view and perpendicular to the shooting angle. Testing was done with several types of noise objects considered in this work namely characters, stamp/company logos, and damages.

<u>Images with Characters</u>

Fig. 12 Container Images with characters (left)original image (right)result

It is demonstrated in Fig. 12 that the algorithm has successfully marked the noises in the container images, for both large – shipping company name- and tiny characters – code and numbers.

<u>Images with Company Logo</u>

Fig. 13 shows results of the algorithm when the input are images with company logo.

Fig. 13 Container Images with Company Sign/Logo (left)original image (right)result

<u>Images with Damages</u>

Fig. 14 demonstrated the algorithm performance with damaged container. It is shown that the algorithm can successfully identified the damage. The damage area is marked with blue circle. It can be seen in the image that the container is dented in some parts near the center of top border. Damage at this level may not be classified as serious damage by some seaports, hence next in line for the subsequent research is to be able to set threshold to conFig. minimum limit for the damage.

Fig. 14 Original container image with damages (top). Damages marked with circle (bottom)

The execution time of this polynomial complexity algorithm is shown to be exceptional. Our experiment with over 10 images yield 2.71 as the average execution time.

VI. Conclusion

In this research, we applied Template Matching Algorithm to detect objects in the side of shipping container. Objects that are considered noise includes but not limited to company name of the cargo container, logo, identification code, signs, labels, and damages. The achieved visual result was satisfactory as well as the computational aspect. The research will not stop at this point. Remained in our list for future work is to distinguish between the identified objects. The ultimate goal is to create a system which can automatically detect whether a container is damaged or not.

I. REFERENCES

[1] W.-C. Lee and C.-H. Chen, "A Fast Template Matching Method for Rotation Invariance Using Two-Stage Process," Kyoto, 2009.

[2] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1042-1052, 1993.

[3] S. Tripathi, V. Sharma and S. Sharma, "Face Detection using Combined Skin Color Detector and Template Matching Method," *International Journal of Computer Applications,* vol. 26, no. 7, 2011.

[4] W.-B. Horng, C.-Y. Chen, Y. Chang and C.-H. Fan, "Driver fatigue detection based on eye tracking and dynamk, template matching," TaipeiI, 2004.

[5] Y. Lee, T. Hara, H. Fujita, S. Itoh and T. Ishigaki, "Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique," *IEEE Transactions on Medical Imaging,* vol. 20, no. 7, pp. 595-604, 2001.

[6] M.-S. Choi and W.-Y. Kim, "A novel two stage template matching method for rotation and illumination invariance," *Pattern Recognition,* vol. 35, no. 1, pp. 119-129, 2002.

[7] G. J. Vanderbrug and A. Rosenfeld, "Two-Stage Template Matching," *IEEE Transactions on Computers,* vol. 26, no. 4, pp. 384-393, 1977.

[8] R. Brunelli and T. Poggiot, "Template Matching: Matched Spatial Filters and Beyond," *Pattern Recognition,* vol. 30, no. 5, pp. 751-768, 1997.

[9] K.-B. Kim, W. Y. Woo and H.-K. Yang, "An Intelligent System for Container Image Recognition Using ART2-Based Self-organizing Supervised Learning Algorithm," in *Simulated Evolution and Learning*, Hefei, 2006.

[10] Lase, "LASE GmbH," [Online]. Available: http://www.lase.de/en/products/port-logistics/lasecdi-container-damage-inspection.html. [Accessed 31 January 2018].

[11] visy.fi, "Visy," [Online]. Available: http://www.visy.fi/products/visy-automatic-container-damage-detection-system/. [Accessed 31 January 2018].

[12] P. M. Mehl, Y.-R. Chen, M. S. Kim and D. E. Chan, "Development of hyperspectral imaging technique for the detection of apple surface defects and contaminations," *Journal of Food Engineering,* pp. 67-81, 2004.

[13] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alexandria Engineering Journal,* 2017.