A Novel Container ISO-Code Recognition Method using Texture Clustering with a Spatial Structure Window

Kyung-mo Koo and Eui-young Cha*

Computer Engineering, Pusan National University,
Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 609-735 Korea
kookyungmo@gmail.com, eycha@pnu.edu

Abstract

In this paper we present a novel container ISO-code recognition method which uses vertical edge information, a spatial structure window, and texture clustering. The vertical edge information is extracted using a top-hat transform. The candidate region and type of ISO-Code is obtained using a Spatial Structure Window (SSW) which wraps around the vertical edges. The ISO-Code is extracted using texture clustering by the K-Means algorithm which is then recognized by a Back-propagation Neural Network (BP). Experiments confirmed the robustness of the recognition algorithm on real images and videos.

Keywords: *Harbor Automation, Unmanned Crane, Container ISO-Code, Texture Clustering, Character Segmentation, Character Recognition*

1. Introduction

As the accelerated development of global industries continues, international trade is playing an important role in national economies. To this end, thousands of containers need to be registered every day at cranes and container terminals. However, until recently, at most trading ports the gates are controlled by a human inspector called the Under-man who interpret and register container information manually. Unfortunately, this not only makes it easy to incur errors, but this paradigm is also too slow to manage the many thousands of cargo containers necessary on a daily basis. Sooner or later a viable automatic method that rapidly identifies containers will need to be implemented.

Many methods have been proposed for container ISO-Code recognition using computer vision and pattern recognition however they are subjected to substantive constraints. Notably, containers in many cases differ in color as well as in layout depending on their owners. Traditional methods of character extraction use vertical edge information. Edge detection and histogram thresholding methods [1] work well with gray level images. Applying a sobel mask to noise-removed images [2], top-hat morphology [3, 4], and alternating horizontal pulses for the edges of the characters in the horizontal direction [5] are alternatives to edge detection and histogram thresholding in several ways but may not be suitable for color images [7]. Oddly, finding ‘U’ character after binarization with 3 threshold levels [6] has been proposed, but the detachable freight containers, trailers and chassis don’t use the product group code ‘U’ [8].

* Corresponding author.
"This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. 2009-0087838)."

We present an overview of our automatic container ISO-Code recognition system used to control the gates and cranes of trading ports in real-time in Section 2, and the spatial structure window and texture clustering method used to extract the ISO-Code in Section 3. How to segment and recognize the ISO-Code image is covered in Sections 4 and 5; the results of testing the algorithm in a real container yard and the evaluation of the efficacy of the device are discussed in Section 6. Section 7 presents our conclusions and lays out our future tasks.

2. The System Overview

An overview of the system is shown in Figure 1. First, the system is triggered by the cameras and movement detectors in order to capture the container images. From these images, it determines the vertical edges, called texture, using white and black top-hat morphology after Gaussian smoothing. Second, it generates 7 types of structure windows. The size of the window is determined for the average size of the character mask. It finds the position and structure of the ISO-Code through the movement of the spatial structure windows. The system then re-arranges each part of the ISO-Code into rows and segments the characters from the re-union image using binarization and K-Means clustering. Finally, the normalized individual characters are recognized using a Back-propagation neural network.

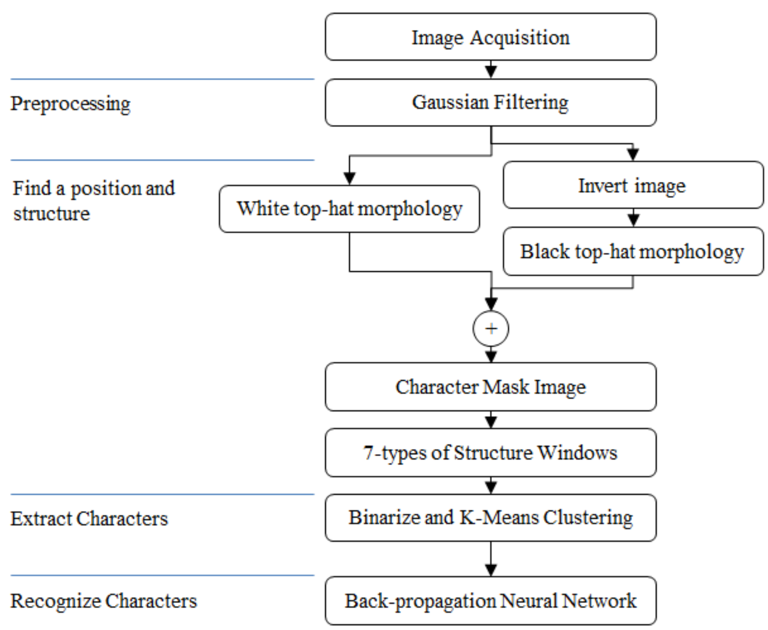


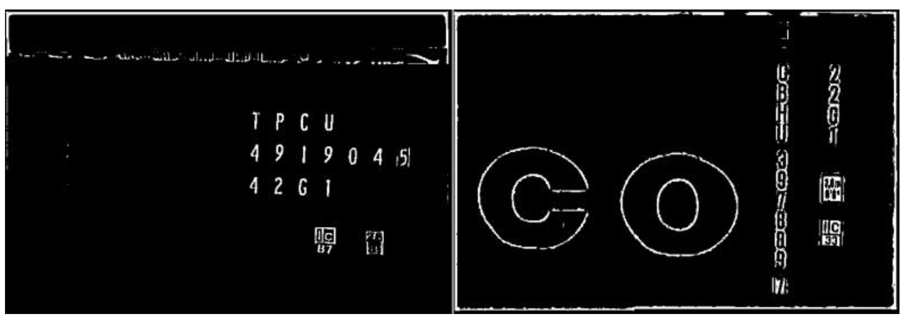
Figure 1. The Container ISO-Code Recognition System

3. Extracting the container ISO-code

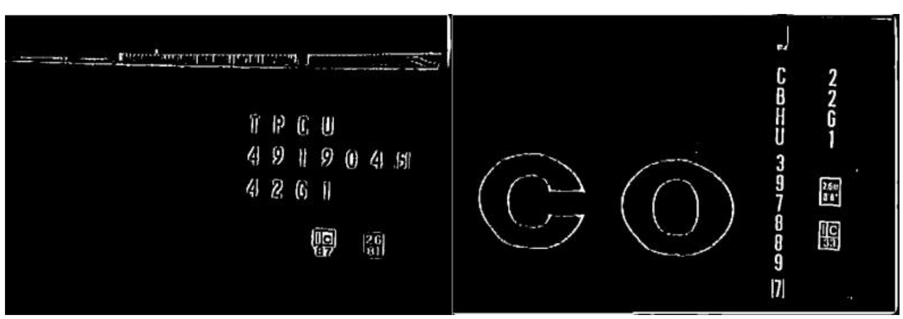
3.1. The Top-hap Morphology

The top-hat morphology (aka top-hat transform) is an operation that extracts small elements and details from the container image. There are two types of top-hat morphology: The white top-hat morphology is defined as the difference between the input image and its opening by the structuring element; it can be used for characters which are brighter than their surroundings. The black top-hat morphology is defined dually as the difference between the closing and the input image in general, but we use opening operation again from an inverted input image instead of the closing. It can be used for characters which are darker than their surroundings. Figure 2 shows examples of top-hat morphology results. Sometimes it is easier to remove the relevant objects rather than finding the irrelevant objects. An important parameter in this morphology is the width of the structuring element. This depends on the thickness of the characters; we use a 31px structuring element in a 720x486px image.

(a) Source images



(b) White top-hat morphology

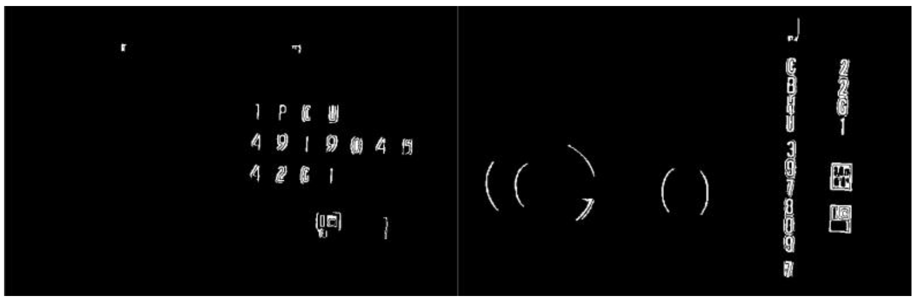


(c) Black top-hat morphology

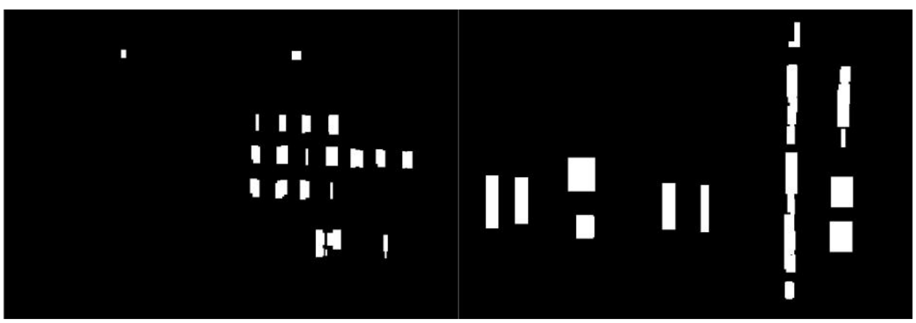
Figure 2. Top-hat Morphology Examples

3.2. The Character Mask

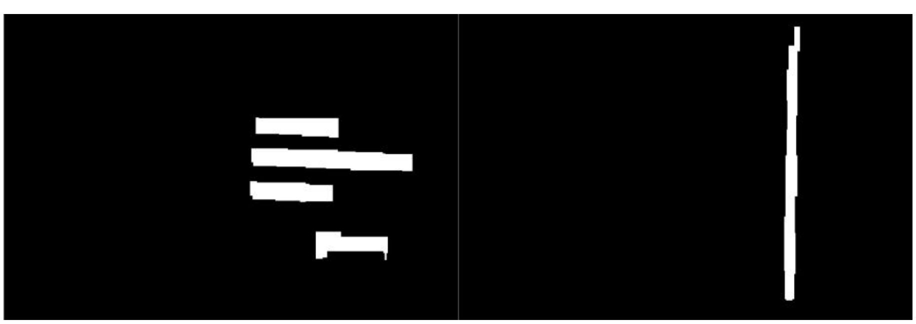
After removing the noise, for instance in some large objects, both images are combined into one, as shown in Figure 3(a) and all of the blobs are filled in using a closing operation, as shown in Figure 3(b). The system then generates a character mask image using the run-length algorithm: If the distances for the objects are larger than 70px in the horizontal direction the objects are combined. We use a 45px distance in the vertical direction. This leaves only the biggest character mask through the labeling and checks its size and ratio. If the height of the mask is larger than 150px and 5 times the width, there is a possibility vertical types are present. If the height of the mask is larger than 250px, it must be a vertical type. Therefore, we do not consider horizontal types in the next step.



(a) Combined images



(b) Maximum images



(c) Character masks

Figure 3. The Character Mask Generation

3.3. The Spatial Structure Window

There are several kinds of container ISO-Code structure types found in real ports, as can be seen in Figure 4. We summarize these types into 5 horizontal and 2 vertical types, as shown in Figure 5. The size of each window is as same as the docking window; its size is decided by the labels which have had a closing operation done on them, in seen in Figure 3(b). The spatial structure window moves on all of the labels with a resizing gap for each window and thereby making a score using its weight. After discriminating all of the labels, the label with the highest score is selected as the correct container ISO-Code type. Figure 6 shows the results of this extraction. After the extraction, the results are combined into one image in order to extract characters through the binarization and K-Means algorithms.

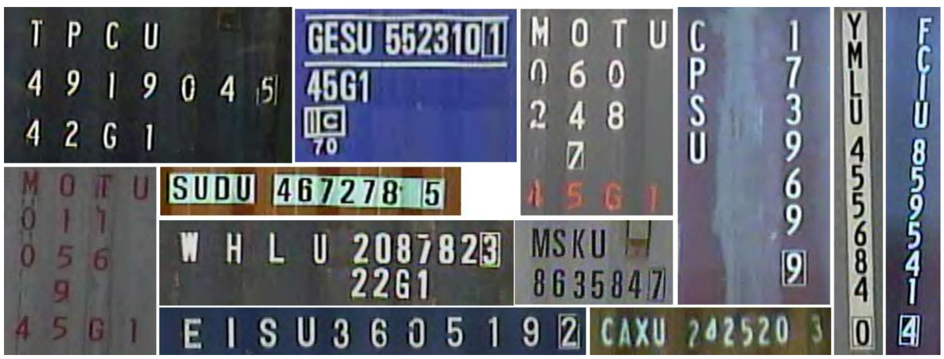


Figure 4. Several Kinds of Container ISO-Code Structures

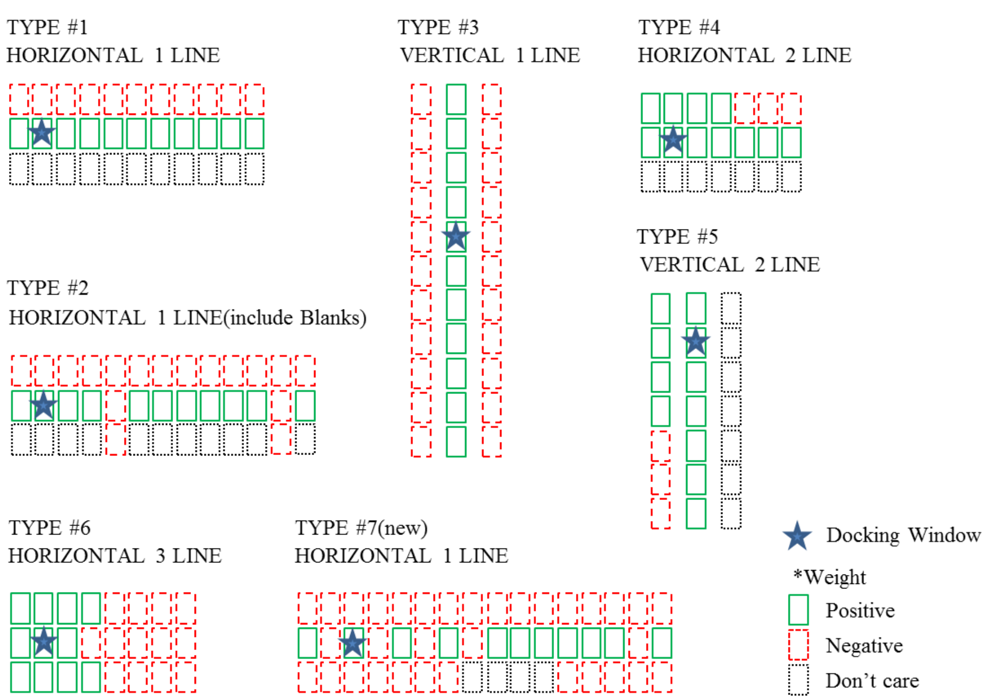


Figure 5. Container ISO-Code Spatial Structure Windows

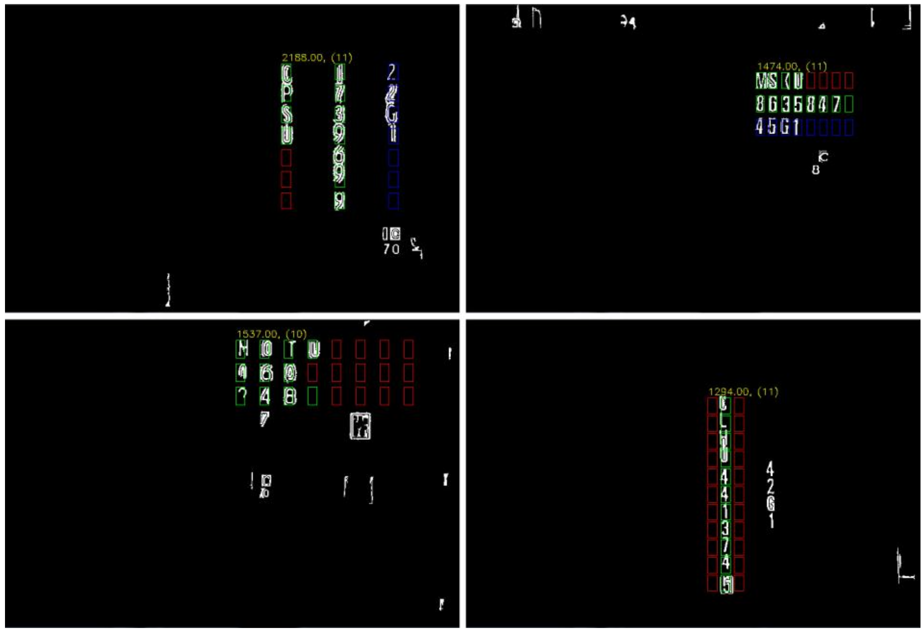


Figure 6. The Container ISO-Code Extraction Results

4. Character Extraction

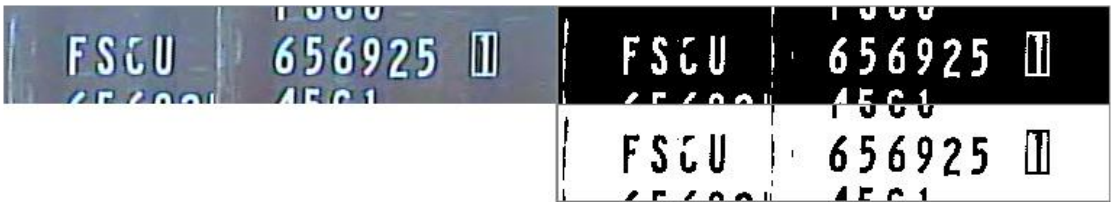
4.1. Binarization

The cropped image and the inversed cropped image are binirized in order to extract the ISO-Code characters using the Otsu method [9]. After both images are labeled, we remove the non-character noise by satisfying the following conditions:

The Container ISO-Code Character Conditions

1. The width of each label must be larger than 2px and smaller than 50px
2. The height of each label must be larger than 15px and smaller than 100px
3. The width to height ratio must be larger than 0.9
4. The height to width ratio must be smaller than 6.0
5. The distance from the center is nearer than half of the height of the cropped image (Use a half of the weight in the vertical structure)

We then divide the characters into two parts: prefixes and numbers. We also check each part for blanks caused by a scratch or a crack. We generate the average distance of the characters and check the distance for every pair. We assume that there is a blank when the distance is greater than a multiple of 1.25. Figure 7 shows the character extraction results.



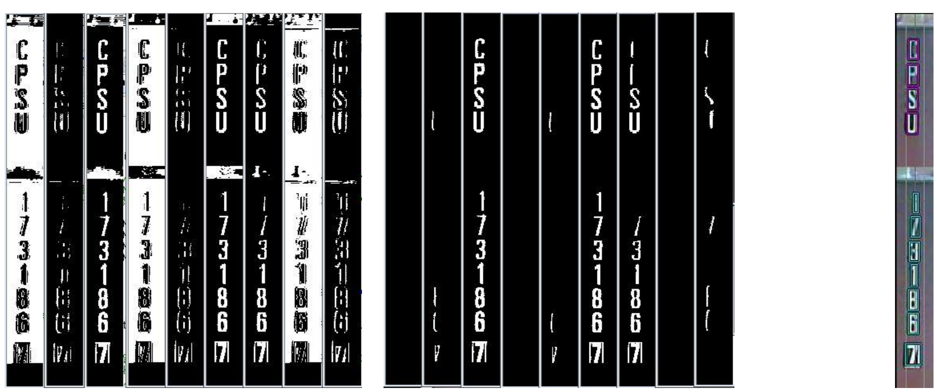
(a) Cropped image (b) Binarized images

(c) Noise reduced images (d) Result with blank

Figure 7. Character Extraction using Binarization

4.2. K-Means Clustering

Binarization is fast however it sometimes may not be able to differentiate between an object and its background because of variations in illumination and noise. We therefore divide the cropped image into three channels. Each image is clustered into three classes using the K-Means Algorithm [10] as shown in Figure 8. After nine images are labeled, we remove the noises in the same manner as binarization.



(a) Clustered images (b) Noise reduced images (c) Extraction result

Figure 8. Character Extraction using K-Means Clustering

5. Character Recognition

The Back-propagation (BP) algorithm is a well-known algorithm used in neural networks. It is efficient in Multi-Layer Perceptron (MLP) which is based on supervised learning. The architecture consists of three layers: the input layer, the hidden layer, and the output layer. The hidden layer is responsible for providing efficiency at the output. The BP algorithm first submits an input for a forward pass through the network. The network output is compared to the desired output and the error for all the neurons in the output layer is calculated. The fundamental idea behind the BP scheme lies in that the error is propagated backward to the earlier layers so that a gradient descent algorithm can be applied. This assumes that the network can 'run backwards', meaning that for any two connected neurons, the 'backward weights' must be the same as the 'forward weights'.

In this paper, we normalize each character image into a 20x30 element image and create an input vector of 600 values. We use 20 hidden nodes and train 200 times in order to reduce the sum squared error to under 0.01. The network can be trained using individual prefixes (alphabet) and numbers (the last number wrapped in a box is included) and each training image is clustered using ART-II before using it as an input vector because the characters can differ in shape (or font) depending on the owner.

6. Experiments

Over 34,000 images were taken by four cameras attached to the side of a crane for a month, as shown in Figure 9(a); these images were recognized through the proposed algorithm in real-time. For the most part, two cameras could take a different side of the same container, as shown in Figure 9(b); we used this advantage in the image acquisition system to increase the recognition rate.

(a) Viewpoints of 4 Cameras (b) Captured Images

Figure 9. The Image Acquisition System using Four Cameras

All individual characters extracted in 18,201 images are recognized well and another 13,455 images are recognized except only one character which is hurt or scratched. These 92.15% of total images are compared with the list which include a load/unload schedule on board, and are matched up the list. Using the list, more than 2000 images can recognize exactly even if only 7 to 9 individual characters extracted. In other words, 98.39% of total image are recognized with the list.

The proposed algorithm takes 240ms on an average to recognize one container in dual core 3GHz and 4GB memory environment and Figure 10 shows the recognition results.

Figure 10. Recognition Results

7. Conclusion

Over In this paper we present a texture clustering method using a Spatial Structure Window to extract container ISO-Codes and their structure. It is an efficient method for automatic character recognition in mass environment images via the BP algorithm.

The proposed method involves an edge detection algorithm using white and black top-hat morphology after Gaussian-smoothing and character segmentation algorithms using binarization and K-Means Clustering. The segmented characters are normalized and recognized using a Back-propagation Neural Network.

The evaluation of the performance of the algorithm was done by conducting field tests in a container terminal. The results yielded a recognition rate of 98.39% for all of the images taken over a month, not only with a very large variability of brightness and contrast, but also in the presence of shadows.

References

[1] R. Ohlander, K. Price and D. R. Pierson, “Picture segmentation using a recursive region splitting method”, In: Computer Graphics and Image Processing, vol. 8, pp. 313-333, **(1978)**.

[2] K. Kwangbaek, K. Sungshin and W. Youngwoon, “An Intelligent System for Container Image Recognition using ART2-based Self-Organizing Supervised Learning Algorithm”, In: LNCS, vol. 4247, pp. 897-904, **(2006)**.

[3] K. Kyungmo, P. Hyunjun, L. Sanglyn and C. Euiyoung, “A Text Extraction in Complex Images using Texture Clustering Method”, In: KIICE, vol. 11, pp. 431-433, **(2007)**.

[4] I. S. Igual, G. A. Garcia and A. P. Jimenez, “Preprocessing and Recognition of Characters in Container Codes”, In: 16th International Conference on Pattern Recognition, vol. 3, pp. 143-146, **(2002)**.

[5] J. C. M. Lee, “Automatic character recognition for moving and stationary vehicles and containers in real-life images”, In: International Joint Conference on Neural Networks, vol. 4, pp. 2824-2828, **(1999)**.

[6] K. Shintaro, M. Kazumasa, T. Mitsuaki, I. Hiroaki and K. Koji, “Development of a container identification mark recognition system”, In: Electronics and Communications in Japan (Part II: Electronics), vol. 87, issue. 12, pp. 38-50, **(2004)**.

[7] S. H. Ong, N. C. Yeo, K. H. Lee, Y. V. Venkatesh and D. M. Cao, “Segmentation of color images using a two-stage self-organizing network”, In: Image and Vision Computing, vol. 20, pp. 279-289, **(2002)**.

[8] CONTAINER Identification system, http://pilhokim.com/images/1/1c/Development_of_a_container_identification_mark_recognition_system.pdf.

[9] N. Otsu, “A threshold selection method from gray-level histogram”, IEEE Transactions on Systems Man Cybernet, SMC-8 pp. 62-66, **(1978)**.

[10] S. Mu-Chun and C. Chien-Hsing, “A modified version of the K-means algorithm with a distance based on cluster symmetry”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, issue 6, pp. 674-680, **(2001)**.

[11] K. Daehee, L. Seungwon and P. Jooki, “Active Shape Model-Based Gait Recognition Using Infrared Images”, IJSIP, Vol. 2, No. 4, pp. 1-12, **(2009)** December.

[12] K. Yong-Cheol, J. Sang-Soo and J. Yong-Kee, “A Tool for Space-Efficient On-the-fly Race Detection, IJDTA, Vol. 4 No. 3, pp. 25-38, **(2011)** September.

[13] N. Yongsik and K. Youngsik, “The Development of Dynamic Brand Equity Chase Model and Its Application to Digital Industry Based on Scanner Data”, IJUNESST, Vol. 2, No. 4, pp. 29-36, **(2009)** December.

Authors

Kyung-mo Koo

He received his master degree in Computer Engineering from Pusan National University in 2005. He is now a PhD candidate in Computer Engineering at Pusan National University. His research interests include image processing, pattern recognition, robot vision, and neural networks.

Eui-young Cha

He received his PhD degree in Computer Science from Seoul National University. He was with the research staff at Korea Institute of Electronic Technology (KIET) and a visiting scholar at London University. He is now a full professor at Pusan National University. His research interests include image processing, pattern recognition, robot vision, and neural networks.