Machine Learning Technology Applied to Production Lines: Image Recognition System

● Tsuyoshi Nagato ● Hiroki Shibuya ● Hiroaki Okamoto ● Tetsuo Koezuka

The recent trend toward mass customization has increased the demand for multiproduct/multivolume production and driven a need for autonomous production systems that can respond quickly to changes on production lines. Production facilities using cameras and robot-based image recognition technologies must also be adaptable to changes in the image-capturing environment and product lots, so technology enabling the prompt generation and well-timed revision of image-processing programs is needed. The development of image recognition systems using machine learning techniques has been progressing with the aim of constructing such autonomous production systems. Furthermore, in addition to the need for automatic generation of image-processing programs, the development of technology for automatically and quickly detecting changes in the production environment to achieve a stable production line has also become an issue. We have developed technology for generating preprocessing programs, extracting image feature values, and optimizing learning parameters and have applied this technology to template matching widely used in image processing and to product accept/reject testing. We have also developed technology for sensing changes in the image-capturing environment by using images captured at the time of learning as reference and detecting changes in subsequent image feature values. These technologies enable the generation of various types of image-processing programs in a short period of time and the detection of signs of change in the image-capturing environment before the recognition rate drops.

1. Introduction

In the field of factory automation, image-processing technology using cameras is applied to a variety of processes such as product manufacturing, assembly, and visual inspection. However, noise and fluctuations in the brightness of captured images frequently occur due to changes in the production environment, which creates a need for robust image-processing programs that can adapt to these changes. Moreover, it is sometimes difficult to use existing programs in the case of startup or upgrade of equipment or change of production lots, so some programs must be regenerated. As a consequence, there is also a need for simplifying the program development process to shorten time-to-market and stabilize production quality.

In addition to the above, the construction of autonomous production systems has been growing in importance to deal promptly with changes and fluctuations on production lines. An autonomous production system using image recognition technology is outlined in **Figure 1**. First, at the time of production-line startup (T_0), an engineer inputs the target data to be recognized in sample images captured by a camera (in this case, the edges in the images to be recognized) to a learning computer and initiates the automatic generation of a complex recognition program. Then, at the time of an environmental change affecting captured images (T_m), the production system itself generates an alarm, and the operator inputs updated target recognition data for sample images taken after the environmental change, thereby initiating an automatic revision to the program. Additionally, in the event of a change in component specifications (T_n), the operator inputs information on that change, thereby triggering automatic regeneration of the program.

In the above way, prompt detection of system changes and revision or regeneration of the recognition

 FUJITSU Sci. Tech. J., Vol. 53, No. 4, pp. 52–58 (July 2017)

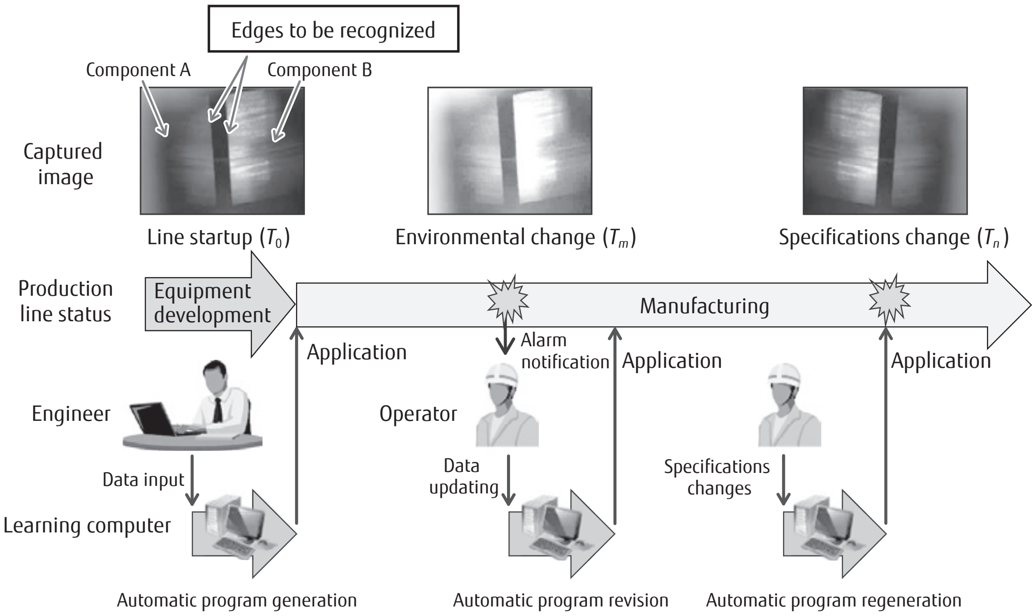


Figure 1
Autonomous production system using image recognition technology.

program makes for stable operation of a production line. Achieving such a production system requires the development of technology for automatic generation of programs and technology for early and automatic detection of environmental changes.

This paper describes technology for automatic generation of image-processing programs and technology for detection of changes in the image-capturing environment to meet this requirement.

2. Technology for automatic generation of image-processing programs

Based on the machine learning technique of genetic programming (GP)[1], a method called "automatic construction of tree-structural image transformation (ACTIT)"[2),3)] has been proposed as a technology for automatically generating image-processing programs. This technology treats an image-processing program as a tree structure consisting of basic image-processing filters and automatically generates the target program by performing combinatorial optimization using GP on a computer.

To date, we have developed GP-based automatic program generation technology and have verified the usefulness of applying it to position recognition processing of geometrical shapes in assembly processing on a production line.[4)] In the following, we describe this technology using two application examples. The first is a method for creating an image-preprocessing step for application to template matching, and the second is technology for extracting image features and optimizing machine learning parameters for application to accept/reject testing in visual inspection processing.

2.1 Application to template matching

Template matching is a process of detecting where a pattern the same as in a previously prepared template image is located within a captured image. It is widely used in mounting inspection and position recognition of production components. Although it is common to apply the same preprocessing to both the captured image and template image to improve recognition accuracy, there are cases in which good results are obtained by applying different processes to each. With this in mind, we first set captured image (cap) and template image (tpl) as two inputs, as shown in **Figure 2 (a)**. Next, we apply different preprocessing functions (C_1–C_5) to each and finally perform matching (M). We define this structure as a tree-structure program of the template matching process and optimize the combination of preprocessing functions by GP.

Optimizing a program by GP requires that we establish an index for evaluating whether the tree-structure program so constructed is working as desired. The output of a program generated by the ACTIT method is only a single image, but in the template matching process, the output consists of information indicating where the template image is positioned within the input image. Thus, in addition to the captured image and template image used as input, we set beforehand the correct position (G_x, G_y) of the template pattern targeted for recognition within the captured image as machine learning data to evaluate the program. In particular, we evaluate the program by comparing the shapes of similarity distributions output after the matching process in addition to using information on the recognized position of the template image. Examples of similarity distributions are shown in **Figure 2 (b)**.

In more detail, we calculate a program evaluation score using two parameters: (1) the shape similarity between similarity distribution $B(x, y)$ output by the program and ideal distribution $T(x, y)$ having a peak only at the correct position and (2) the difference between the recognized position and the correct position. In this way, a program that outputs similarity distribution $L(x, y)$ having a peak only in the vicinity of the correct position obtains a high evaluation score, thereby enabling the automatic generation of a program with good performance for the template matching process.

A validation experiment demonstrated that this technology can automatically generate a program that can recognize the correct position of a template pattern after about two hours of learning without the risk of erroneous recognition even if the captured image has other areas similar in shape to the template image targeted for recognition. Computer specifications and learning conditions used in the experiment are listed in **Table 1**. Comparison of the performance of this technology with that of an image-recognition program generated by a production-equipment developer that

Table 1
Computer specifications and learning conditions.

Computer specifications	
OS	Windows 7 (64 bit)
CPU	Intel Xeon Processor 2.4 GHz
Memory	64 GB (8 GB×8)
Learning conditions	
Population	30
Number of generations	500
Crossover probability	1.0
Mutation probability	0.9

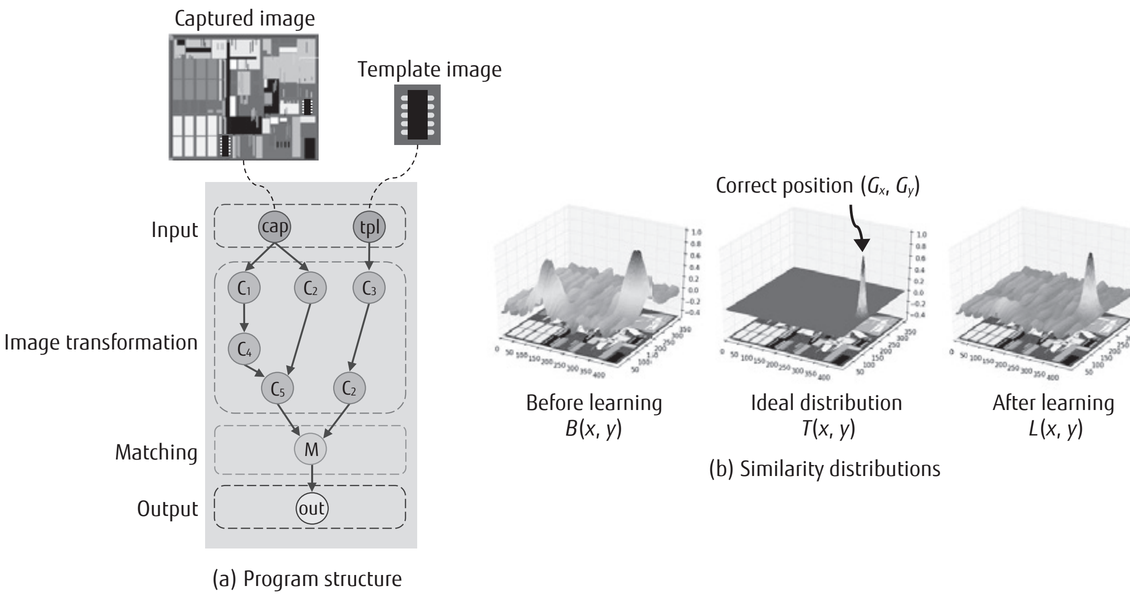


Figure 2
Application to template matching.

required about one week of tuning revealed that the proposed technology achieved equivalent performance and is therefore an effective approach to program generation.

2.2 Application to accept/reject testing

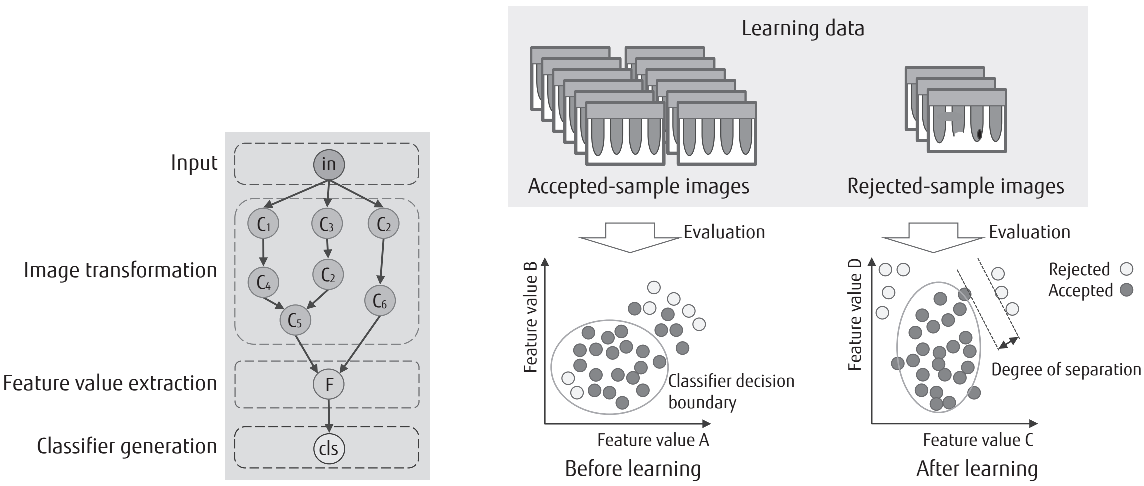
It is common on production lines to judge the presence of a component or the acceptability of component mounting by image recognition. To achieve such accept/reject testing using images, the selection of image feature values for making judgments and a method for constructing a classifier (decision rule) using those feature values are important.[5] However, combining the feature values of all images can result in a massive search space and an impractical learning time. In addition, rejection during production is usually rare, so it is common to be constrained by the requirement of having to use a very small number of images for rejected samples in training and in generating a classifier.

As shown in **Figure 3 (a)**, we first define the series of processes from image transformation (C_1–C_6) to feature value extraction (F) and parameter determination for classifier generation (cls) as a tree-structure program. We then automatically generate a program for accept/reject testing by optimizing the defined series by GP. Additionally, to perform machine learning using a small number of images for rejected samples as described above, we use an accuracy rate based on cross-validation as a program evaluation index. We also use degree of separation based on the distribution of feature values as another evaluation criterion with the aim of generating a program with high generality (having predictive ability with respect to new data).

Examples of learning data and decision boundaries of automatically generated classifiers are shown in **Figure 3 (b)** (for explanatory purposes, the dimension of feature values is set to 2). In this way, simply inputting learning data (images) and instructing which of those images are acceptable and which are unacceptable makes it possible to simultaneously optimize and obtain an image-feature extraction method and classifier-construction method. This enables accurate accept/reject testing as reflected by distribution after learning in comparison with distribution before learning.

In a validation experiment, we performed learning using images of 20 samples (16 accepted; 4 rejected) for a period of approximately two hours (using the same computer as that for which the specifications are listed in Table 1 and under the same learning conditions, except for the population being set to 100). The accuracy rate for the subsequent accept/reject testing was greater than 99% for 300 images.

As described above, we have shown that the automatic generation of image-processing programs by GP can be applied to a variety of image-processing



(a) Program structure (b) Examples of learning data and classifier decision boundaries

Figure 3
Application to accept/reject testing.

techniques used on production lines and that program generation time can be shortened for production line startups and changeovers (i.e., when the production line setup is changed due to a change in the type of product or work).

3. Technology for detecting changes in image-capturing environment

When executing an image-processing program generated by GP, its recognition accuracy may drop due to changes in the image-capturing environment or specifications of the target object compared with those at learning time. Consequently, if it is determined at some point that the image-processing accuracy required for production cannot be maintained, it would normally be necessary to add input images based on the image-capturing environment and target object at that time to the original learning data and retrain the program. This, however, would require that production-line operation be suspended until a new image-processing program is generated. Such downtime can be prevented by detecting signs of environmental changes early and performing retraining before the recognition rate of the image-processing program drops noticeably.

Our technology calculates image feature values from captured images and identifies changes in the image-capturing environment from the fluctuation in feature values with respect to the sample images used at learning time.

3.1 Image feature values

It is not uncommon on a production line for the area illuminated by lighting and the camera installation position to become misaligned due to line modifications such as replacement of production equipment or adjustment of hardware components. This problem can be dealt with by assuming that such changes in the image-capturing environment will appear as changes in the brightness, focus, etc. of the captured images and that such changes can be grasped from image feature values such as average luminance, contrast, and spatial frequency.[6]

The feature value distributions of images captured during two different time periods for one set of production equipment and of images captured during one of those time periods for another set of production equipment are shown in **Figure 4**. The feature space in the figure consists of three main feature values (brightness, high frequency, and angle) condensed from the image feature values described above by main-component analysis. Each one corresponds to an axis. These results show that feature value distributions can differ due to temporal differences (equipment set A between 2013 and 2014) and equipment differences (between equipment sets A and B in 2014). On the basis of these findings, we decided to detect changes in the image-capturing environment using the three main image feature values described above.

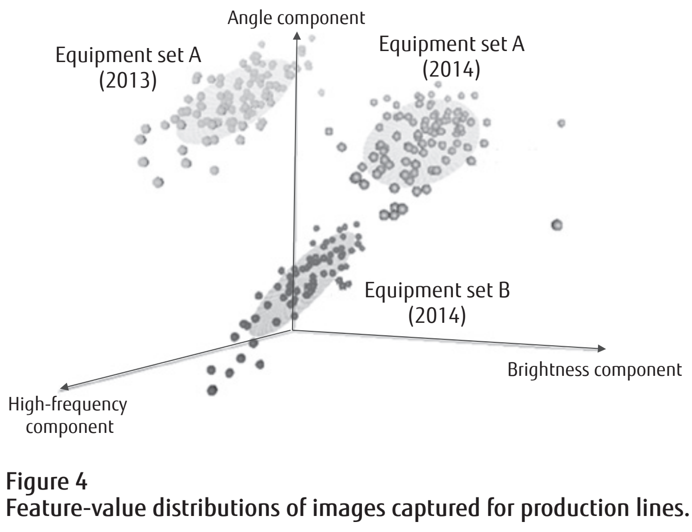


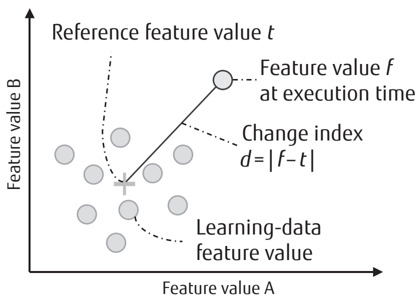
Figure 4
Feature-value distributions of images captured for production lines.

3.2 Evaluation by distance in feature space

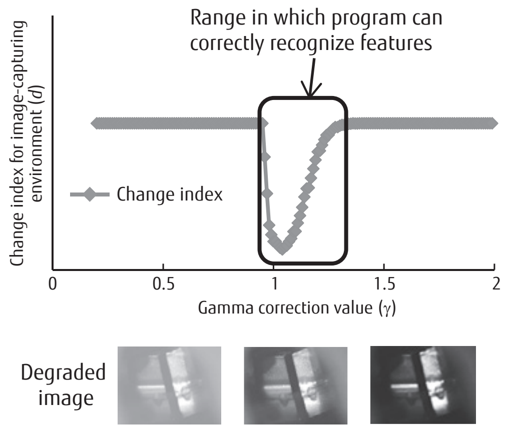
A change in the image-capturing environment is detected by using the image feature values at the time of equipment startup (time of program generation) as reference and calculating and checking the distance between that reference value and feature values every time the program is executed. An overview of the change index for an image-capturing environment is shown in **Figure 5 (a)** (for explanatory purposes, the dimension of feature values is set to 2). First, the image feature values are calculated for each image used as learning data at the time of equipment startup, and the center of that feature value distribution is set as reference feature value t. Then, at the time of program execution, distance d between feature value f of a captured image and reference feature value t is calculated, and that value is treated as the change index for the image-capturing environment.

To test the validity of this change index, we created captured images that reproduce degradation in brightness and focus, as shown in **Figures 5 (b)** and **(c)**, and checked the transition in change index d in relation to the amount of degradation. Brightness is changed by gamma correction processing (image contrast increases for a gamma correction value γ of 1 or greater and decreases for γ less than 1) in relation to a sample image at the time of equipment startup. Defocus is produced by a smoothing process (an image becomes increasingly blurred as defocus intensity σ increases). In addition, we performed actual processing on these degraded captured images by using an automatically generated program and checked recognition performance. As shown in Figures 5 (b) and (c), the change index increased as the captured images became increasingly degraded. We also found that such changes can be detected while the program is still correctly recognizing features. This means that a change in the image-capturing environment can be detected before the recognition rate of the image processing program drops noticeably.

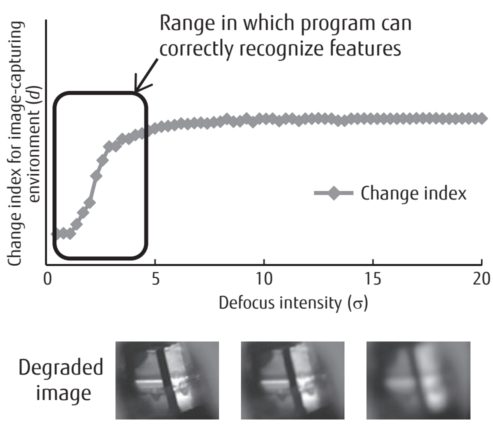
As described above, changes in the image-capturing environment that do not yet affect program



(a) Overview of change index for image-capturing environment



(b) Transition in d versus change in brightness



(c) Transition in d versus change in focus

Figure 5
Change index for image-capturing environment.

performance can be detected from image feature values. In other words, retraining can be executed before a drop in recognition performance of image processing occurs. This ability to respond promptly to changes in the production environment enables stable operation of the production line.

4. Conclusion

We described technology for automatically generating image-processing programs for application to template matching and accept/reject testing. The technology for template matching was able to generate a program in approximately two hours, and its performance was equivalent to that of one generated by an equipment developer. The technology for accept/reject testing was able to generate a program having an accuracy rate greater than 99% for 300 test images.

Our performance evaluation experiment demonstrated that our technology for detecting changes in the image-capturing environment using image feature values can detect such changes before the recognition rate of the image-processing program drops noticeably. This capability will contribute to stable operation of production lines. Going forward, our plan is to construct and test an autonomous production system by applying the technologies introduced in this paper to a variety of image recognition processes.

References

1) J. R. Koza: Genetic Programming, Cambridge. MIT Press, 1992.
2) S. Aoki et al.: Automatic Construction Method of Tree-structural Image Conversion Method ACTIT. Journal of the Institute of Image Information and Television Engineers, Vol. 53, No. 6, pp. 888–894, 1999 (in Japanese).
3) T. Nagao: Evolutionary Image Processing. Shokodo, 2002 (in Japanese).
4) T. Nagato et al.: Automated Design of Image-Processing Programs for Positioning Parts. 2014 Japan Society for Precision Engineering Autumn Meeting Proceedings. Tottori, Japan Society for Precision Engineering, 2014, pp. 151–152 (in Japanese).
5) C. M. Bishop: Pattern Recognition and Machine Learning. Springer, 2006.
6) K. Kushima et al.: Content-based Image Retrieval Techniques based on Image Features. IPSJ Journal, Vol. 40, No. SIG3 (TOD1), pp. 171–184, 1999 (in Japanese).

Tsuyoshi Nagato
Fujitsu Laboratories Ltd.
Mr. Nagato is currently engaged in the research of image-processing and image-recognition systems for factory automation applications.

Hiroki Shibuya
Fujitsu Ltd.
Mr. Shibuya is currently engaged in the development of factory automation technologies.

Hiroaki Okamoto
Fujitsu Laboratories Ltd.
Mr. Okamoto is currently engaged in the research of image-processing and image-recognition systems for factory automation applications.

Tetsuo Koezuka
Fujitsu Laboratories Ltd.
Mr. Koezuka is currently engaged in the research of image-processing and image-recognition systems for factory automation applications.