Automatic Recognition of Wire Bobbins using Machine Vision Techniques

머신 비젼 기술을 이용한 전선 보빈의 자동인식

Tai-Hoon Cho
(조 태 훈)

요 약 : 이 논문은 에나멜 전선의 제조공정의 자동화에 있어서 핵심역할을 하는 보빈의 자동인식을 위한 머신 비젼 시스템에 관한 것이다. 이 시스템의 역할은 컨베이어 라인의 팔레트 위에 놓인 보빈들의 영상을 CCD 카메라로 취득, 분석하여 보빈 형태, 색상, 제조공정번호 등의 다양한 정보를 추출하여, 전체 생산공정을 제어하는 주 컴퓨터로 보내는 일을 수행한다. 이 비젼 시스템은 개발된 후 에나멜 전선 생산공장에 설치되어 일정 시험기간을 거쳐 현재 성공적으로 운영되고 있다.

Keywords : machine vision, inspection, automation, object recognition, robot vision

I. Introduction

Machine vision has been applied to a number of industrial applications for quality control and improvement that is critical in the manufacturing process. Detailed reviews of automated visual inspection are discussed in survey papers, for example, [1][2]. In this paper, a machine vision system is described that plays a key role in an automated handling system of the process of manufacturing enamel wire. The main motivations for automation of manufacturing enamel wire come from both productivity increase and difficulty in hiring workers for production lines. It is so hot and noisy inside an enamel wire manufacturing factory that very few are willing to work in such working conditions.

In enamel wire manufacturing process, enamel wires are produced and automatically wound on bobbins by machines. Every fully wound bobbin has a manufacturing label attached on its top by a worker, and then it is placed on a pallet moving in free-flow conveyor lines. On the label there is a bar code representing manufacturing information for a specific bobbin. Since the label can be placed randomly on the top of a bobbin, it is impossible to read bar codes using a conventional bar code reader. The only way for reading bar codes placed in a random orientation and position is to use a machine vision system. The function of this system is to acquire images of bobbins placed on pallets in a conveyor line using CCD cameras and to analyze them to extract various information such as bobbin types, colors, and numeric manufacturing codes. This information is sent to a main host computer that controls the whole enamel wire manufacturing process. In what follows, system requirements for the machine vision system are described.

접수일자 : 1998. 1. 26., 수정완료 : 1998. 7. 10.
조태훈 : 한국기술교육대학교 컴퓨터공학과

Bobbins are placed on pallets and moved in a free-flow conveyor line. Each line produces about 400 wound bobbins per day, so the line flow is relatively slow. There are two inspection stages that should be simultaneously controlled by the machine vision system. The first stage, bobbin type recognition stage, inspects whether or not a bobbin is placed on an incoming pallet. If a bobbin is found in the pallet, this stage recognizes its type and decides whether enamel wire is wound on the bobbin. It should handle nine types of bobbin. Depending on its type, the height of a bobbin is ranged from 160 mm to 630 mm while the diameter of the top side of a bobbin is between 123 mm and 380 mm.

The second stage, label recognition stage, recognizes both the color of a bobbin and a numeral manufacturing code inside a label that is arbitrarily attached on the top of a bobbin. The color of a bobbin is light yellow, gray, or black. A label is rectangular and has information printed in black onto white background. It is surrounded by black lines with thickness of 1.5 mm. It is assumed that all labels should be placed within a circle that has a radius of 160 mm and a center equal to the center of the pallet. Of course, the position and orientation of labels can be random inside this circle. The manufacturing code of a label consists of 9 numerals without any space. Each numeral of a code in each label is 4 mm wide and 6 mm high. The space between consecutive numerals is 1.5 mm wide.

In both stages, bobbins are kept momentarily still by a mechanical stopper during image acquisition. Recognition results obtained at each stage are sent to a main host computer via a RS-232C serial port. Processing time requirement is less than 5 and 20 seconds for the bobbin type recognition and label recognition, respectively. Error rate required is less than 1 % error rate for both stages. Most importantly, this system should be extremely robust in order to operate 24 hours per day.

In the following sections, configuration of the overall system, algorithms used in the software implementation, performance of the system implemented and deployed, and finally a conclusion are presented in some detail.

II. System configuration

Fig. 1 shows a functional block diagram of the whole system. The bobbin type recognition stage inspects a bobbin from the side using back lighting because side view of each type of bobbin has unique shape characteristics. A silhouette image of the side of each bobbin is obtained by a B/W camera with a fixed focal length lens and analyzed to determine its existence and type. The label recognition stage inspects each bobbin from the top using front diffused lighting and two B/W cameras (with different fields of view) mounted on a X-Y moving robot. The main reason for using two cameras and a X-Y robot is that the field of view required is too large for a camera.

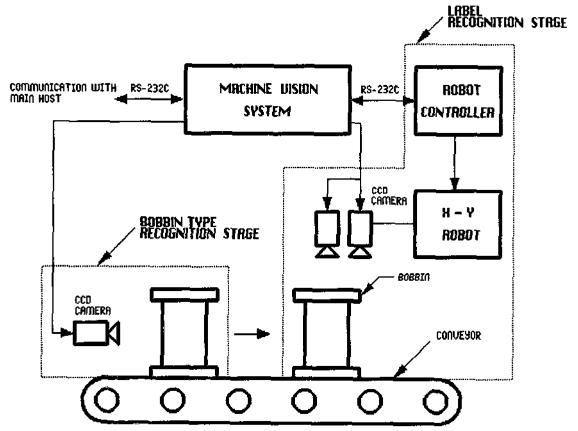


Fig. 1. A functional block diagram of the whole system.

In Fig. 2 is shown a block diagram of the machine vision system. Its hardware consists of a VME rack, a hard disk, an image display monitor, a text monitor, a keyboard, and a mouse. There are three boards in the VME rack. The first one is a CPU board using a 33 MHz 68040 CPU with 4 serial ports. These serial ports (RS-232C) are connected to a text monitor, a mouse, a main host computer, and a robot controller. The second one is a frame grabber that can digitize upto 4 camera video inputs into 512x480 images. The last one is an output board that can control an alarm system. The alarm system is needed because the machine vision system is typically operated without attendance of an operator. If anything goes wrong with the system operation, an alarm light and a buzzer are turned on so that nearby workers can notice it. The output board also has a function to reset the robot. This is very useful for resuming normal operation without an operator's intervention in case a system error related with the robot system occurs. (Actually, in the field there occurred a robot error approximately twice a week in 24-hour-per-day operation due to possibly high electrical noise spikes.) A hard disk is served as a saving place of a training data and images. All the system hardware is enclosed in a system rack with an air-conditioning system. The air-conditioner keeps the temperature inside the rack around 30 ℃ because the temperature inside the factory is too high in summer (typically around 40 ℃). For bobbin type recognition, a 2/3 inch B/W camera with a lens of focal length f=12.5 mm inspects bobbins from the side using back lighting. The bobbin is illuminated from its backside to the camera. To expand the illumination over the camera's field of view, a diffuser is placed between the bobbin and a light source. As the light source, an array of fluorescent lamps was used. For label recognition, two 2/3 inch B/W cameras were used. One, Camera 1, is used to acquire an image to locate the label center position with a lens of f=25mm. The other camera, Camera 2, is used for label code recognition with a lens of f=50mm. Note that each field of view area in vertical direction is set to approximately to 340mm and 160 mm for Camera 1 and 2, respectively. These cameras were mounted to a X-Y robot that first moves Camera 1 to the pallet center and then Camera 2 to the label center position. The position control of the X-Y robot is done via a serial port by the CPU board.

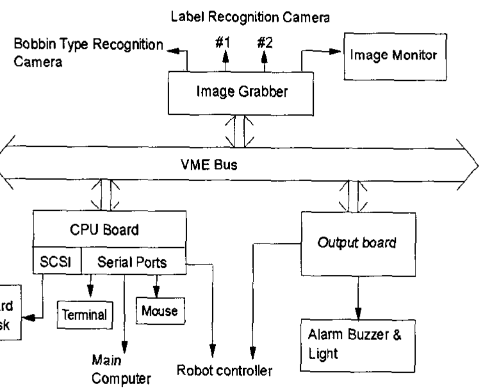


Fig. 2. A block diagram of the machine vision system.

III. Algorithms

Software of this system was all written in C. It is run under a real-time OS (Operating System) on the CPU board. Algorithms used in bobbin type recognition and label recognition are described below in some detail.

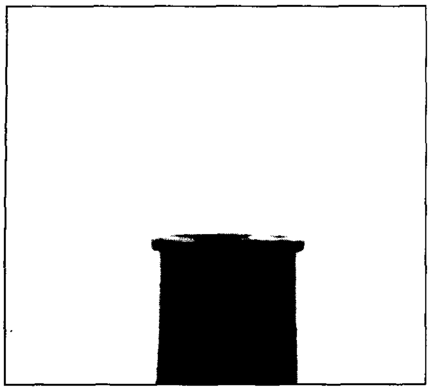


Fig. 3. A side view image of a bobbin acquired in bobbin type recognition stage.

Fig. 3 shows an input image acquired in the bobbin type recognition stage. At this stage, existence of a bobbin in the pallet is easily determined by analyzing horizontal projection of an input image gray-level data. The type of a bobbin is found by determining its height and the diameter of its top surface and comparing with previously trained values for each bobbin type. The height of a bobbin is determined based on horizontal projection of the image gray-levels. The diameter is determined by using vertical gray-level projection of a window image that covers the top side of a bobbin. Lastly, whether or not a bobbin is wound with enamel wire can be decided by measuring the diameter of its body. The body diameter of a fully wound bobbin should be much larger than that of a bare bobbin without wire.

Fig. 4. An image obtained by camera 1 in the label recognition stage.

The label recognition stage begins with identifying the color of a bobbin. The color of a bobbin is classified into one of three different colors (light yellow, gray, and black) based on the gray level information of the top of the bobbin. Experimentally, it was found that only gray-level information is necessary to discriminate among these colors because each color has a unique gray-level histogram peak of its own. Gray levels at four sample points that are evenly distributed around the bobbin center on the top surface are used to determine the bobbin's color. The recognition of a manufacturing code on a label can largely be divided into two procedures: label region extraction followed by label code recognition. Label region extraction is proceeded as the following steps.

Step 1 : Camera 1 is moved to the center of the pallet, an image is captured using Camera 1. Fig. 4 shows an input image that contains top side of a yellow bobbin.

Step 2 : Connected component labeling [3] is performed to find connected regions after binarization using well-known Otsu's automatic thresholding method [4].

Step 3 : Hole filling operation is done to the connected regions. This operation is necessary because many holes are generated after the connected component labeling due to characters inside the label region.

Step 4 : Among the regions, a region R is chosen that is most similar to the label based on both area and compactness. (Compactness is defined as 4π (area)/(perimeter)2, where perimeter is the length of the region boundary). Then the center position (Cx, Cy) of the label region R is calculated using

$$(Cx, Cy) = \left(\frac{1}{N} \sum_{(x,y)\in R} x, \frac{1}{N} \sum_{(x,y)\in R} y\right) \tag{1}$$

where N is the area of the label region R and $(x,y) \in R$ represents the x-y coordinates of a pixel in R. The center position of the label extracted is shown in Fig. 5.

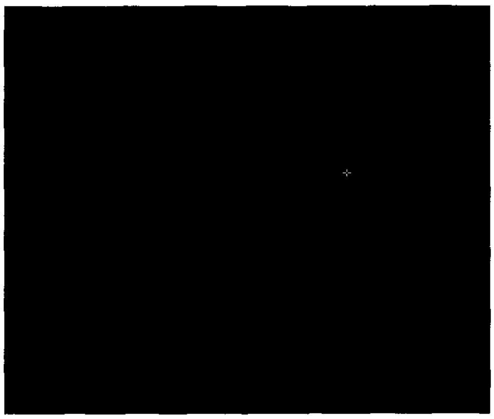


Fig. 5. Label center location extracted.

After the label center position is found, label code recognition is performed as follows.

Step 1 : The X-Y moving robot moves Camera 2 to the center of the label and a second image as shown in Fig. 6 is obtained. Processing steps similar to those of the label region extraction are performed to extract the label area in the second image.

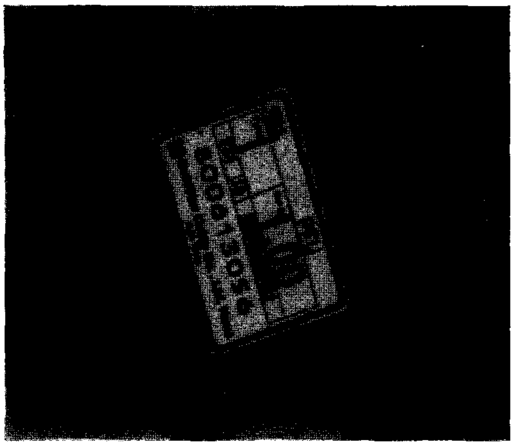


Fig. 6. A label image obtained by camera 2.

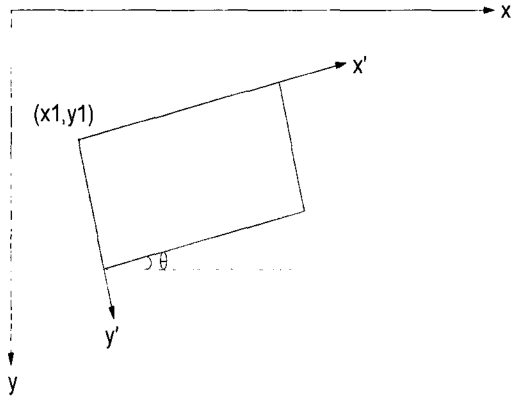


Fig. 7. Coordinate convention used in image rotation.

Step 2 : The center (Cx, Cy) and rotation angle θ of the label region R are calculated. The center of R is extracted again as described above to obtain a more accurate position. Rotation angle θ of the label region R is found by

$$\theta = \frac{1}{2}\tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right) \tag{2}$$

where μ_{pq} is pqth order central moment given by

$$\mu_{pq} = \sum_{(x,y)\in R}\sum (x-Cx)^p (y-Cy)^q \tag{3}$$

Step 3 : The center and rotation angle of the label are used together with structural information of the label to determine where the manufacturing code exists. (Structural information of the label such as layout is already known). Once determined, this code area is then rotated by θ so that a numeral code appears upright. Bi-linear interpolation [5] is used to yield a rotated image of the code area of the input image. Following the convention shown in Fig. 7, an original coordinates system (x,y) is associated with a rotated coordinates system (x',y') by

$$x = x'\cos\theta + y'\sin\theta + x_1, \tag{4}$$

$$y = -x'\sin\theta + y'\cos\theta + y_1. \tag{5}$$

Then $g(x',y')$, the gray level at (x',y'), based on bi-linear interpolation is given by

$$\begin{aligned} g(x',y') = &(1-\alpha)(1-\beta)f(\lfloor x \rfloor, \lfloor y \rfloor) \\ &+(1-\alpha)\beta f(\lfloor x \rfloor, \lfloor y \rfloor+1) \\ &+\alpha(1-\beta)f(\lfloor x \rfloor+1, \lfloor y \rfloor) \\ &+\alpha\beta f(\lfloor x \rfloor+1, \lfloor y \rfloor+1), \end{aligned} \tag{6}$$

where $\alpha = x - \lfloor x \rfloor$, $\beta = y - \lfloor y \rfloor$, and $f(x,y)$ denotes the gray level at (x,y) in the original image. $\lfloor x \rfloor$ represents the maximum integer not exceeding x.

Step 4 : To recognize the numeral code in the rotated code area, two main algorithms were used. The first algorithm is to segment the code area into each consecutive numeral area using both vertical and horizontal projection of the gray-level data of a sub-image corresponding to the code area. Fig. 8 shows an example of this segmentation. Notice that peaks of the horizontal projection data correspond to points dividing consecutive numerals. After the segmentation process, each numeral area is matched using a normalized correlation method against model templates of numerals for recognition. The normalized correlation $R(x,y)$ is calculated by

$$R(x,y) = \frac{(N\sum_i I_i M_i - (\sum_i I_i)(\sum_i M_i))}{\sqrt{(N\sum_i I_i^2 - (\sum_i I_i)^2)(N\sum_i M_i^2 - (\sum_i M_i)^2)}} \tag{7}$$

where I and M denotes an input image and a model template image, respectively, and index i covers all positions of the template M and N represents the number of pixels in M. The value of R is always in the range -1 to 1, inclusive. A value of 1 signifies a "perfect match" between the image and the model. A value of -1 signifies a "perfect mismatch"; that is, the feature found in the image is the negative of the model. The negative of a model or image is a corresponding image in which the sense of light and dark has been reversed. The model templates for numerals (0, 1, 2, ..., and 9) are previously stored into a disk in the training process. Each numeral area is recognized as number i if the normalized correlation with model template of number i yields a maximum value among the normalized correlations with all templates of numbers and if the correlation value exceeds a certain confidence level. Fig. 8 shows an example of the numeral code recognition using the first algorithm. This algorithm works well in normal cases where orientation of the label is accurately extracted and the label area is clean. However, if the rotation of the label cannot be calculated accurately for any reason, or if there is some dirt in the area between consecutive numerals of the label, the segmentation process could fail. To remedy for this situation, the second algorithm was devised. The second algorithm, first of all, tries to find and recognize the first numeral from the leftmost of the code area to the right using the normal correlation. Once the first numeral has been found and recognized, the second numeral is searched and found using the normalized correlation within a certain area around its expected center position. The third numeral's position is estimated based on the first and the second numerals already found. Within a window around this estimated position, the third numeral is searched using the normalized correlation. This process is repeated until all the numerals are found and recognized. Note that the second algorithm contains no segmentation process and this is applied only when the first algorithm fails to recognize numerals.

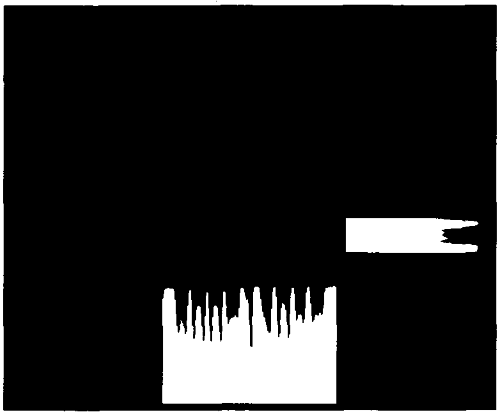


Fig. 8. A final recognition output.

Ⅳ. Performance

The vision system implemented was tested for three months in an enamel wire manufacturing factory. During the test period, the algorithms were continuously refined for achieving maximum performance of the machine vision system. After rigorous testing, the normal operation of the whole automated production lines including the vision system began. Currently, two machine vision systems are in use in two different production lines in the factory. About 400 bobbins per day were processed by each of the machine vision systems. The performance of the machine vision system was very satisfactory. The bobbin type recognition stage performed almost perfectly, and the label recognition stage performed very well with virtually zero error rate and reject rate less than 0.5%. The final testing of three weeks rejected 83 bobbins out of 17419 bobbins, yielding a reject rate of 0.48%. The reasons for the rejection include label impairment, lifted label, badly printed code in the label, algorithm imperfection, etc. Rejections due to label impairment and lifted label occupied approximately 60% of the total rejections. Average processing time for each bobbin is about 0.5 and 15 seconds for bobbin type recognition and label code recognition, respectively.

Ⅴ. Conclusion

In this paper, a machine vision system for automatic recognition of bobbins was presented that plays a key role in an automated handling system of the process of manufacturing enamel wires. The function of this system is to acquire images of bobbins placed on pallets in a free-flow conveyor line using CCD cameras and to analyze them to extract various information such as bobbin types, colors, and numeric manufacturing codes. This information is sent to a main host computer that controls the whole manufacturing process. This vision system was deployed and tested in an enamel wire manufacturing factory with great success.

References

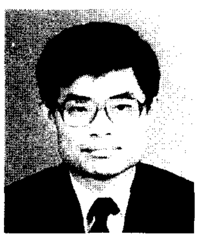
[1] R. T. Chin, "Survey : automated visual inspection : 1981 to 1987," *CVGIP* 41, pp. 346-381, 1988.

[2] T. S. Newman and A. K. Jain, "A survey of automated visual inspection," *Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 231-262, 1995.

[3] Y. Shirai, *Three Dimensional Computer Vision*, Springer-Verlag, 1987.

[4] N. Otsu, "A threshold selection method from gray level histograms," *IEEE Trans. Syst. Man Cybernet.*, vol. 9, pp. 62-66, 1979.

[5] A. Rosenfeld and A. Kak, *Digital Picture Processing*, Academic Press, 2nd ed., 1982.



조 태 훈

1981년 서울대학교 전자공학과 졸업. 1983년 한국과학기술원 전기 및 전자공학과 석사. 1991년 Virginia Polytechnic Institute & State Univ. 공학박사. 1992년 3월~1998년 2월 LG산전(주) 연구소 책임/수석 연구원 1998년 3월~현재 한국기술교육대학교 컴퓨터공학과 조교수. 관심분야는 머신비전(Machine Vision), 컴퓨터비전, 영상처리.