論 文

# IC핀 조사를 위한 시각 조사 방법

正會員 崔 聖 鎭* 正會員 趙 東 來** 正會員 金 榮 海*
正會員 羅 克 煥*** 正會員 金 春 吉****

# Visual Inspection Method by Pyramid Data Structure

Seong Jhin CHOI*, Dong Rae CHO**, Young Hae KIM*,
Keuk Hwan RA***, Chun Kil KIM**** *Regular Members*

**要 約** 시각조사의 일은 대부분 숙련된 조사자에 의해 이루어지므로 시간이 경과 할수록 조사의 질은 감소한다. 그러므로 조사과정의 자동화가 요구되고 있으며, 실용적인 시각조사시스템을 위해서는 실시간에서의 데이터 처리능력, 다양한 입력물체에 대한 유연성 및 저렴한 시설비가 요구된다.

본 논문에서는 이러한 요구에 적합한 조사방법을 제안한다. 기존의 하드웨어 방식과는 달리 소프트웨어에 의해 조사데이터를 처리함으로써 조사시스템을 유연성 있게 하기 위하여 피라미드 데이터구조를 이용한 divide-and- conquer 기술과 간소화된 패턴 매칭방법을 결합한 시각조사 알고리즘을 제시한다. 이 방법에 의해 비교되는 패턴의 데이터 수를 감소시킬 수 있었고, 그 결과 조사속도를 줄일 수 있었다.

***ABSTRACT*** For a visual inspection system to be viable, it must in general achieve the same or lower costs as those incurred by manual methods, it must have a reasonable inspection rate, and it must be reliable and maintainable : for the visual inspection system, inspection at high speed, flexibility for variety of products, and low cost are needed.

In order to solve these problems, we propose the visual inspection algorithm by divide-and- conquer technique using the pyramid data structure and the development of simplified pattern matching method. By this method we can reduce the number of data required to compare with patterns and make the inspection time short.

## I. Introduction

During the past two decades, pattern recognition and image processing have witnessed rapid growth in both methodology and their applications to real world problems[1-4]. On the other hand, advances in sensor technology provide better and cheaper image recording and analysis equipment[5-6]. The cost of using computers per unit time is also decreasing rapidly. But visual inspections in most manufacturing processes have depended mainly on human inspectors. However, the labor of inspection is hard, and the quality of inspection depends on the inspector's skill and inevitably

* 仁德工業專門大學 電子科
Dept. of Elec. Eng. Induk Institute of Design
** 國防科學硏究所
*** 光云大學校 電子工學科
Dept. of Telecommunication Engineering
Kwang Woon Univ.
**** 韓國科學技術大學
論文番號：90－39(接受1990. 1. 31)

declines with each inspection job over time. Therefore, to automate such a labor intensive inspection task by means of pattern recognition techniques is the motivation of this paper.

For a visual inspection system to be viable, it must in general achieve the same or lower costs as those incurred by manual methods, it must have a reasonable inspection rate, and it must be reliable and maintainable ; i. e., for the visual inspection system, inspection at high speed, flexibility for variety of products, and low cost are needed[7].

In order to solve these problems, we propose the visual inspection algorithm by the development of simplified pattern matching method and divide-and- conquer technique using the pyramid data structure.

The techniques used in automatic visual inspection fall into the three broad categories discussed below.

(1). The image of the object being considered is compared to the defect free image of the object, pixel by pixel. If the gray level difference for any two pixels from the two images exceeds a threshold, then a defect exists at that pixel. This method is widely used in industrial inspection systems because it is simple, fast and reliable[8-9]. However, pixel by pixel comparison requires the storage of complete defect free images.

(2). A set of features extracted from a defect free image of the object is stored. The same features are extracted from the image of the object being inspected and are compared to the former set to determine whether the object is defective or not[8-9]. In this case, the numerical specification of various features often represents a satisfactory characterization of the product under inspection. But when the product to be inspected is very complex, simple and powerful features cannot be found esaily or clearly defined.

(3). A small window is moved on the sensed image. If the image inside the current window area violates some generic properties of the product, the product is defective. The generic properties are inferred from a defect free porduct. This technique is successful only when the inspection criteria can be transformed into a set of rules that can be applied uniformly throughout the entire image.

As mentioned in the above part, the current inspection system using pattern matching was not suited to a variety of objects. So we selected the inspection method using pattern matching processing and judgement processing by software. But if pattern matching between the object pattern and the reference pattern is performed over the whole pattern, it takes time for matching processing. By this reason, the matching method by software mentioned above cannot be applied to a practical inspection system.

In order to realize high speed, we apply divide-and- conqure technique by the pyramid data structure to the reference pattern and object pattern and match corresponding pixels on these patterns. By this method, we can reduce the number of data required to compare with them and make the inspection time short.

An image pyramid is a widely used model for visual perception. Pyramid architectures for image processing and analysis have been of interest to many researchers [10]. A pyramid is a stack of cellular arrays of decreasing size. It is generated by a bottom-up process which takes a compact region on the lower level, called a kernel, and applies a local function to this set of pixels to derive the value of a cell in the next higher level.

Kelly[11] was the first researcher to use a form of the pyramid data structure in image processing. His task was to extract the outline of a human head from a digitized image. He decided that his task would be easier if he first extracted the outline in a coarse image and used this result as a "plan" to find the more detailed outline in the original image. Uhr[12] was the next to use the pyramid. He had his own name for it. He applied it to the general task of pattern recognition. His contribution in this regard was to discuss the possibilities for developing systems which characterized images through successive layers of the pyramid. Chen and Pavlidis[13], klinger and Dyer[14] investigated image segmentation by texture using pyramid data structure. Tanimoto[15] has investigated the ramifications of distortions due to averaging (i.e. building the layers of the pyramid). He gave quantitative answers to questions about the robustness of edges and corners. Rosenfeld[16] investigated a generalization of the pyramid linking approach. Shneire[17 18] described a pyramid linking method of compact region extraction.

Previous paper using pyramid data strcuture has considered numerous methods for linking pixels at successive levels based on their similarity of property values. In this paper, we describe an algorithm for computing statistical information about the input image using the standard nonoverlapped pyramid. The proposed method is easier and simplier than the previous method using linking concept.

The automatic assembly of visual inspection system is another area that is worthwhile for investigation. A method for automatic alignment has been proposed by the authors [19]. Therefore this topic will not be discussed in this paper.

## Ⅱ. The pyramid architecture

Before describing hierarchical matching technique using pyramid data structures, we define pyramid. A pyramid strcuture is shown in figure 1.

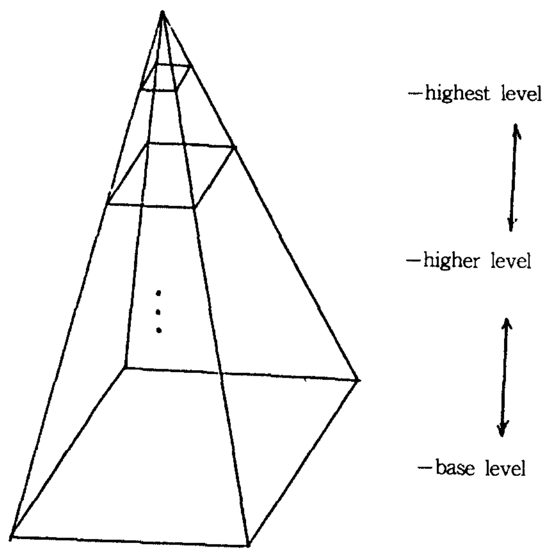



Fig. 1 Image Pyramid.

A pyramid is an exponentially tapering stack of square arrays of processors which from now on we call nodes. The arrays are called the levels of the pyramid; the largest array, level 0, is called the base. Each node is connected to its neighbors (north, south, east and west) on its own level provided they exist. In addition, each node N above the base is connected to a block of nodes on the level below it; these nodes are called N's children and N is called their parent. For simplicity, we assume that the blocks all have the same size and shape (except possibly at the edges of the arrays), so that every node has the same number c of children. A node can have more than one parent and it always has at least one except if it is on the hightest level. Again, we assume that every node has the same number p of parents. To insure exponential tapering, we require that c is a multiple of p, say $c=rp$ where $r > 1$ ; thus the number of nodes decreases by a factor of r from level to level.

As a simple example of a pyramid, let the levels have sizes $2^n \times 2^n$, $2^{n-1} \times 2^{n-1}, \cdots$, and let the children of node (i, j) on the level k be cells (2i, 2j), (2i+1, 2j), (2i, 2j+1), and (2i +1, 2j+1) on level k-1. We call these the southwest, southeast, northwest, northeast children, respectively. In this case each node has only one parent, and r=4. We call this example the standard nonoverlapped pyramid, since in this case the blocks of children do not overlap. Another simple example is the case where each node above the base has a $4 \times 4$ block of children, and these blocks overlap by 50% in both the i and j coordinate directions; this implies that each node has four parents, so that here too r=4. We call this example the standard overlapped pyramid.

When we process an image using a pyramid, we input the image to the base of the pyramid, one pixel per node. We can imagine that each node in the base incorporates a photosensor, and that the base lies in the image plane of a camera. We shall think of the processing as consisting of a series of synchronized steps, where at each step every cell obtains information from its children, neighbors, and parents and performs some computation on this information. In this algorithm that we will describe, at a given step only the nodes on one level of the pyramid are actively computing; the others are inactive.

Our algorithm is designed to be implemented on a pyramid, but it can also be implemented on a number of other computational structures. One example is a prism[10], which is a stack of n+1 arrays all of the same size $2^n$ by $2^n$. Here node(i, j) on level k is connected to its siblings and also to nodes (i, j), $(i+2^k, j)$, and $(i, j+2^k)$ on level k+1. Each computational step on a pyramid can be simulated by a few steps of computation on a prism. Another important example is a hypercube having $2^n$ nodes. We assign to each node a distinct n-bit binary number, and connect each node N to the n nodes whose numbers differ from N's in exactly one binary place[20].

The architectures considered in this paper are regularly structured; for example, in our pyramids, the blocks of children are all congruent (e.g., $4 \times 4$) and are regularly spaced (e.g., their centers are distance 2 apart horizontally and vertically).

## Ⅲ. Statistical computation

In this section we describe an algorithm for computing statistical information about the imput image, using the standard nonoverlapped pyramid. In this pyramid each node N on level k has a block of 4 children on level k-1, a block of 4 grandchildren on level k-2,$\cdots$, and a block of $2^k \times 2^k$ descendants on level 0 ; we call this last block N's block of the image. To compute statistical information about the blocks of size $2^k \times 2^k$, our algorithm performs k computational stages, each of which involves only a bounded amount of computation. In this sense, we can say that the total amount of computation is proportional to the logarithm of the block size.

Our algorithm can operate either on the original input image, or on images generated

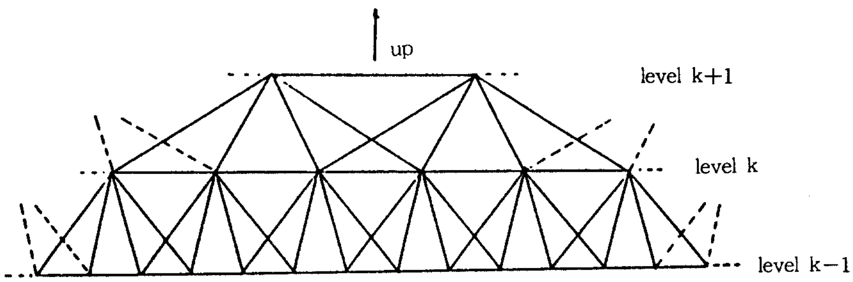



Fig. 2 One-dimensional relation in a pyramid levels.

from the input image using local operations in which the new value of a pixel depends only on the old values of that pixel and a set of its neighbors. Evidently any local operation can be performed on the input image, using the neighbor connections in the base of the pyramid, in a bounded number of steps.

1) summing and mean

Each node computes the sum of the values in its block of the images. Initially, the value of a node on level 0 is its input value. At the first step of the algorithm, each node on level 1 adds the values of its four children; at the second step, each node on level 2 adds the values of its four childern; and so on. Evidently, after k steps, the nodes on level k have computed the sums of the values in their $2^k \times 2^k$ image blocks. This style of computation is called divide and conquer; the task of adding $4^k$ numbers is recursively decomposed into k subtasks – i, e., a number of subtasks proportional to $\log(4^k)$ – each of which involves adding only four numbers so that each subtask involves only a bounded amount of compuatation. Each node at level k can compute the average pixel value in its image block by adding the values, as in 1), and dividing the sum by $4^k$. Note that the nodes at level k have thus computed a reduced resolution version of the image.

## Ⅳ. Image registration and pattern matching

A description of the pyramid structure is in order before we proceed pttern matching. Every pixel has one father ond four sons, see Figure 2. Each level of the pyramid has a local coordinate system. If the base of the pyramid is 256 by 256 pixels, for example, then its x and y coordinates on level 0 of pyramid are in the range 0 to 255. At level 1 of the pyramid the x and y coordinates range from 0 to 127. A pixel on level k of the pyramid with local coordinates (p, q) will be represented by pixel(k, p, q). This will uniquely identify every pixel of the pyramid.

To make the pyramid for pattern registration, the following values need to be computed and stored for each pixel as follow.

$$\mathrm{val}(k,\ p,\ q)=\mathrm{floor}\,(\frac{1}{4}\ \sum_{x=0}^{1} 0 \sum_{y=0}^{1} \mathrm{val}(k+1),2p+x,\ 2q+y))$$

The pyramid algorithm is given below;

for each pixel(x, y, z) of the pyramid from the bottom up :
begin

```
if(a=0)
  then pixel(x, y, z). graylevel=initialpic
  (x,y)
if(a>0)
  then pixel(x, y, z). graylevel=0
  and for each son(i, j) of pixel(x, y, z)
    begin
      pixel(x, y, z). graylevel= pixel(x,
      y, z). graylevel+ (son(i, j). ginfrayle-
      vel) / 4
    end
end
```

This procedure is easily seen to be 0(log n), for an n by n input image.

The coarsest level of the reference pattern and test pattern is formed by the above procedure and pattern matching between the reduced patterns is outputted as follow:

$$Y_1=\overline{X(a_1)\oplus X(a'_1)}$$

$$Y_2=\overline{X(a_2)\oplus X(a'_2)}$$

$$Y_n=\overline{X(a_n)\oplus X(a'_n)}$$

where $a_n$ and $a'_n$ are positions of pixels extracted from inspection areas on the reference pattern and those on the object pattern respectively, X mean the binary data of the each pixel obtained by thresholding.

## V. The detected object delineation

The method of retrieving the detected part projects the object back onto the base of the pyramid. A pixel compares his initialized gray level value with that of his fathers' to determine which of them is closest. The pyramid has initialized each pixel by a simple block averaging process. A pixel is then picked on some level k of the pyramid that represents an object below. We consider only the sub pyramid that has this pixel's father at its apex. The other sons of this father are labelled as background pixels. On each successive level every pixel that lies in this sub-pyramid takes on the label of the father that minimizes the graylevel difference between them. The graylevel values that are compared are those computed when the pyramid was initialized. This method is a labelling process and the algorithm is given below.

```
pick a pixel(k, p, q) to expand.
At every pixel(x, y, z) of the pyramid
from the top level.
bigin
  if(pixel(x, y, z)=cell(k, p, q))
    then label pixel(x, y, z) as an object
    pixel.
  if(pixel(x, y, z)=brother (pixel(k, p,
    q)))
    then label pixel(x, y, z) as a
    background pixel.
  if(h_father (pixel(x, y, z), object) or
    h_father (pixel(x, y, z), backgraound))
    then if(clocest_ father(pixel (x, y, z
          ))=object)
          then lable pixel(x, y, z) an
          object pixel.
          else label pixel(x, y, z) a
          background pixel.
end
```

## VI. Experiment

The algorthm just described was applied to the photograph image of 14 pin IC shown in Fig. 3. This image is 256 by 256 pixels (8 bit / pixel); thus the top (2×2) level of the

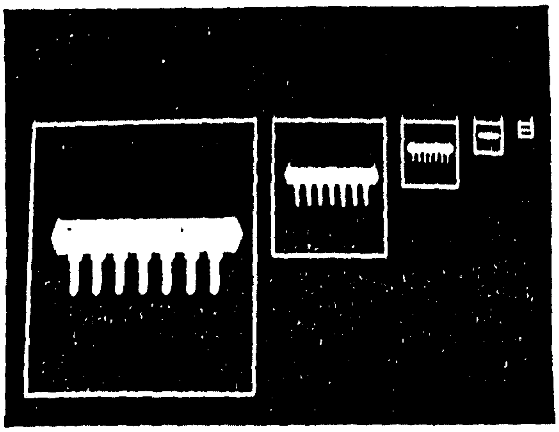


Fig. 3 Fig. 4

Fig. 3 Reference Image(256 by 256 pixels)
Fig. 4 Reduced reference image for matching.

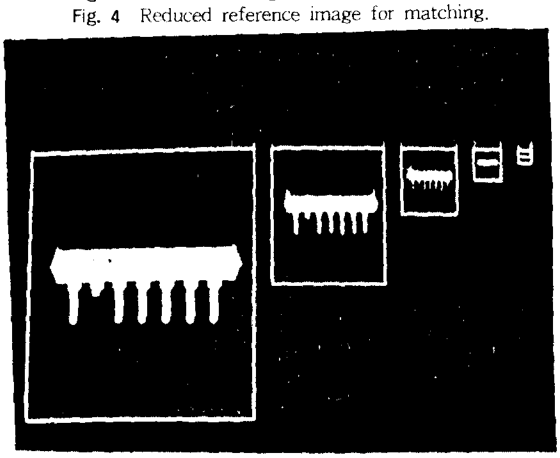


Fig. 5 Fig. 6

Fig. 5 Defective test image (256 by 256 pixels)
Fig. 6 Reduced defective image for matching

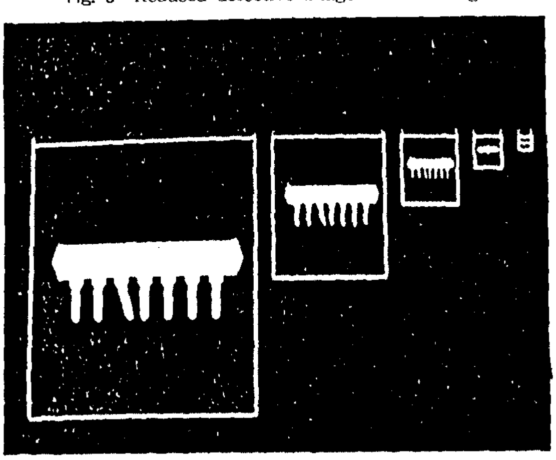


Fig. 7 Fig. 8

Fig. 7 Defective test image (256 by 256 pixels)
Fig. 8 Reduced test image for matching.

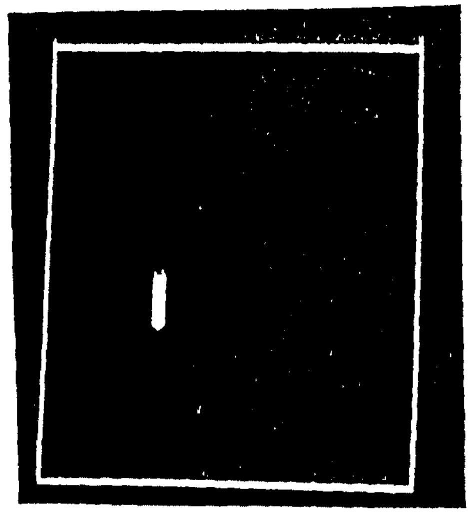


Fig. 9

Fig. 9 Defective pattern of Fig. 5 obtained by the proposed method

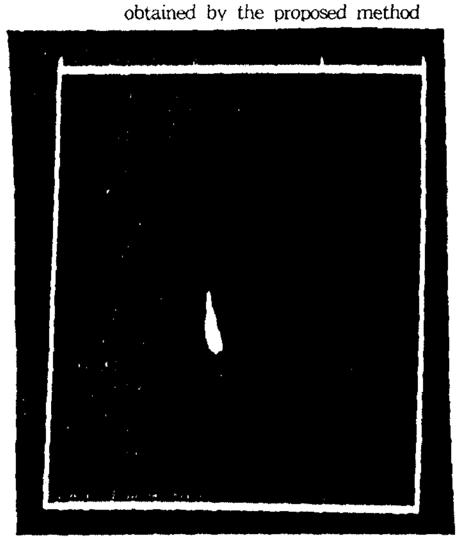


Fig. 10

Fig. 10 Defective pattern of Fig. 7 obtained by the proposed method.

pyramid is level 7. But starting level s is determined by the inherent levels of interest for the picture class. Typically we have

$$s=(2L/3)$$

where L is the number of levels in the pyramid. Therefore in this paper, starting level is level 4 (16×16). Fig. 4 shows the reference pyramid data structure obtained by the mean algorithm for computing statistical information about the input image Fig. 3. Fig. 5, 7 show the defective object images with 256 by 256 pixels and Fig. 6, 8 are the pyramid data structure of test images. Fig. 9, 10 show an output images of defective part obtained by matching between the reduced reference image and test image and processing for the delineation of detected part.

The conventional method (brute force method) is a reasonable approach to consider when ease of implementation or understandability is a criterion for judgement. Two drawbacks soon become obvious. The first is that large amounts of time are required to execute the exhaustive matching. The second is that the matching may not be successful because of the noise present. Therefore, the conventional method evaluates the pattern matching at each of the pixels of the original picture thereby requiring 65536 times for 256 by 256 picture and 256 times in this method.

The method described in this paper could be implemented very efficiently on a cellular pyramid machine[10]. Here the local process at each level could be carried out in parallel in 0(constant) time per iteration, so that the total computation time would be proportional to the number of levels, i.e. o(log n) for n by n image. The results are quite reasonable for a process that can be implemented in such a short computation time.

Even when implemented on a sequential machine, the pyramid approach has advantages with respect to computational cost as indicated in above section.

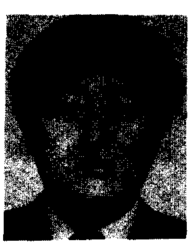
## Ⅶ. Conclusion

The conventional inspection system using pattern matching was not suited to a variety of objects. So we selected the inspection method using software for all inspection processing. But if pattern matching between the object pattern and the reference pattern is performed over the whole pattern, it takes time for matching processing. By this reason, the matching method by software mentioned above cannot be applied to a practical inspection system. In order to realize· high speed, we proposed the visual inspection algorithm by divide and conquer technique using the pyramid data structure and the development of simplified pattern matching method. By this method we can reduce the number of data required to compare with patterns and make the inspection time short. The top level of the pyramid will be 1 by 1 pixel image by the proposed method but strating level s is determined by the inherent levels of interest for the picture class. In this paper, by many experiment we have s=(2L/3) where L, is the number of levels in the pyramid. Therefore starting level s is level 4(16×16 rsolution); that is, 256 pixels at level 4 are matched to be inspected, pattern matching time takes about 1 second, which is fast enough for practical inspection.

## References

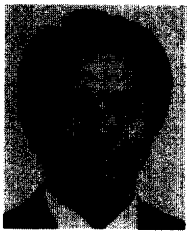

1. I. Alexander, Digital Systems for Industrial Automation, Crane-Rusak, New York, 1981.
2. J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, 1982.
3. B. K. P. Horn, Robot Vision, MIT Press, New York, 1986.
4. K. S. Fu, Robotics : Control, Sensing, Vision, and Intelligence, McGraw-Hill, New York, 1987.
5. R. J. Sanderson, "A survey of the robotics vision

industry", Rob. World, pp. 22~31 Feb. 1983.
6. A. Rosenfeld, "Machine vision for industry : Tasks, tools, and techniques", Image Vision Comput., 3, No. 3, pp. 122~135, 1985.
7. T. Nakajima, "A simplified visual inspection system using a personal computer", Journal of Robotic System, 2(3), pp. 253~264, 1985.
8. K. S. Fu, "Pattern recognition for automation visual inspection", Proc. SPIE, Robot Vision, 336, pp. 12~19, 1982.
9. K. S. Fu, "Pattern recognition for automatic visual inspection", Computer 15, No. 12, pp. 34~40, 1982.
10. A. Rosenfeld, Multiresolution Image Processing and Analysis, Springer Verlag, Berlin, 1984.
11. M. D. Kelly, "Edge detection in pictures by computer using planning", Meltzer and Michie, eds., Machine Intelligence 6, Edinburgh University Press, Edinburgh, Scotland, 1971.
12. L. Uhr, Layered, "Recognition Cone Network that Preprocess, Classify and Described", IEEE Trans. Computer, C-21, July, pp. 758~768, 1972
13. P. C. Chen and T. Pavldis, "Image segmentation as an estimation problem", CGIP 4, pp. 104~119, 1975.
14. A. Klinger and C. R. Dyer, "Experiments on picture representation using regular decomposition", CGIP 5, pp. 68~105, 1976.
15. S. L. Tanimoto, "Pictorial feature distortion in a pyramid", CGIP 5, pp. 333~352, 1976.
16. A. Rosenfeld, "Segmentation and estimation of image region properties through cooperative hierarchical computation", IEEE Trans. Syst., Man, Cybern., vol. SMC-11, pp. 802~809, 1981.
17. M. Shneire, "Using pyramids to define local thresholds for blob detection", IEEE Trans. PAMI-5, pp. 345~349, May, 1983.
18. M. Shneier, "Extracting compact objects using linked pyramids", IEEE Trans. PAMI-6, pp. 229~237, March, 1984.
19. O. Silven and H. Hakalahit, "A method for aligning printed circuit boards with design data", in proceedings, 4th Scandinavian Conf. Image Analysis, Trodheim, Norway, June, 1985.
20. M. Spann, "A quad-tree approach to image segmentation which combines statistical and spatial information", Pattern Recog., 18, pp. 257~268. 1985.

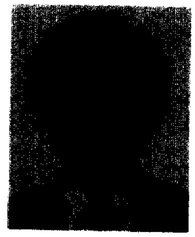


**崔 聖 鎭**(Seong Jhin CHOI) 正會員
1960年 1月 1日生
1982年 2月 : 光云大學校 電子工學科 卒業(工學士)
1984年 2月 : 光云大學校 大學院 電子工學科 卒業(工學碩士)
1984年 8月~現在 : 光云大學校 大 學院 電子工學科(博士課程修了)
1984年 8月~1985年 2月 : 韓國特殊專門要員
1987年 3月~現在 : 仁德工業專門大學 電子科 助教授

**趙 東 來**(Dong Rae CHO) 正會員
1950年 9月23日生
1973年 2月 : 延世大學校 電子工學科 卒業(工學士)
1979年 2月 : 延世大學校 大學院 電子工學科 卒業(工學碩士)
現在 : 延世大學校 大學院 電子工學科 (博士課程 修了)
現在 : 國防科學研究所 研究室長



**金 榮 海(Young Hae KIM)** 正會員
1943年 5月 4日生
1969年 2月：光云大學校 應用電子工學科 卒業(工學士)
1982年 2月：建國大學校 大學院 電氣·電子工學科 卒業(工學碩士)
1969年 3月～1974年 2月：光云電子工業高等學校 教師
1974年 3月～1982年 2月：金烏工業高等學校 教師
1982年 3月～現在：仁德工業專門大學 副教授



**羅 克 煥(Keuk-Hwan Ra)** 正會員
1950年10月13日生
1973年 2月：延世大學校 電子工學科 卒業
1977年 2月：延世大學校 大學院 電子工學科 卒業
1977年：亞州大學校 專任助教
1978年～1981年：불란서 ENSEEIHT 에서 工學博士 取得
1981年～1986年：光云大學校 電子科 副教授
1987(1年)：美國COMSAT研究所 芳問教授
1988年～現在：光云大學校 電子工學科 在職中

**金 春 吉(Chun Kil KIM)** 正會員
• 現在：韓國科學技術大學 教授