International Journal of Fuzzy Logic and Intelligent Systems, vol. 1, no. 1, June 2001 pp. 13-23

A Knowledge-Based Machine Vision System for Automated Industrial Web Inspection

Tai-Hoon Cho, Young-Kee Jung and Hyun-Chan Cho

School of Information Technology, Korea University of Technology and Education

Abstract

Most current machine vision systems for industrial inspection were developed with one specific task in mind. Hence, these systems are inflexible in the sense that they cannot easily be adapted to other applications. In this paper, a general vision system framework has been developed that can be easily adapted to a variety of industrial web inspection problems. The objective of this system is to automatically locate and identify "defects" on the surface of the material being inspected. This framework is designed to be robust, to be flexible, and to be as computationally simple as possible. To assure robustness this framework employs a combined strategy of top-down and bottom-up control, hierarchical defect models, and uncertain reasoning methods. To make this framework flexible, a modular Blackboard framework is employed. To minimize computational complexity the system incorporates a simple multi-thresholding segmentation scheme, a fuzzy logic focus of attention mechanism for scene analysis operations, and a partitioning of knowledge that allows concurrent parallel processing during recognition.

Key Words : machine vision, industrial web inspection, modular Blackboard framework, multi-thresholding segmentation scheme, fuzzy logic focus of attention mechanism.

I. Introduction

The new world economy is going to mean increased competition among companies in various areas, especially in manufacturing industries. This competition among companies is going to put more pressure on company management to find more cost effective methods for increasing productivity and improving quality control. Machine vision systems provide a mechanism to accomplish both of these objectives. While humans perform well on most visual inspection problems, their performance is susceptible to a number of factors. Therefore, the automation of this visual inspection task should improve quality control. Machine vision systems can also be used to increase productivity

An extensive survey of machine vision systems for automated industrial inspection is given in [1, 2]. Machine vision technology has been successfully applied to a number of industrial inspection problems. To be industrially useful, an industrial inspection system must be able to meet the following requirements [3].

1. Speed. The operation speed of the vision system must match that of the production line.
2. Accuracy. The accuracy of the vision system in defect detection must be high enough to be acceptable.
3. Flexibility. The vision system must be flexible enough to accommodate changes in products and competent enough to inspect parts in an uncontrolled environment.

Unfortunately, most current vision systems for automated industrial inspection have a number of marked limitations [3]. Most current vision systems for automated industrial inspection are very inflexible. They cannot be easily adapted to work on a problem other than the very specific application for which they were designed. Many of the limitations of the current machine vision systems have resulted from the need for real-time processing. Another cause is the need for robustness. Researchers have not been able to create truly robust general-purpose computer vision methodologies. There have been only a few attempts to create fairly general-purpose vision systems, e.g., [4, 5].

The first problem of industrial machine vision systems should be mitigated by current improvements in computational capabilities of microprocessors. This development should motivate the rapid growth in the industrial machine vision area. However, if there is to be rapid growth, one cannot start from scratch in developing each new machine vision system. One needs to have at least a general framework from which to begin. This leads to the second problem that remains a major point of concern. This paper primarily attempts to address the second problem, i.e., the robustness issue, in

Maniscript received March 15, 2001; revised May 1, 2001.

industrial web inspection, which is a rather general and very important subclass of industrial machine vision problems.

Industrial web inspection problems involve inspection of planar (flat) material or patterns. These typically have the following characteristics:

Surface of the material is relatively flat.

Material flow is continuous.

Flow rate is relatively high to high (typically, 2 - 20 linear feet per second).

Relatively high spatial resolution is required (up to 100 pixels per inch).

Defects make up small percentage of total surface area.

The same defect type manifests itself in many different ways.

This industrial web inspection is important because most industrial inspection problems except inspection of volumetric shaped materials fall into this category [3]. Problems of inspecting printed circuit boards (PCBs), microcircuit photomasks, integrated circuits (ICs), paper, textiles, surfaces of lumber, etc. are included in the web inspection area.

In this paper, a flexible computer vision system framework is described that attempts to address a fairly general class of industrial inspection problems, the web inspection problem. The purpose of the vision system is to locate and identify "defects" on the surface of the material being inspected.

The design criteria used to create this vision system framework are as follows.

1. The vision system must be robust. It must be able to accommodate without adjustment small changes in lighting and/or changes in the surface characteristics of the material caused by acceptable variations in processing.
2. The vision system must be flexible in order to enable the system to be easily adapted to a number of web inspection problems.
3. The vision system must be able to process high spatial resolution image data. In general, high resolution imagery is required to detect small defects, although exact resolution required depends on the application being considered.
4. The operation of the vision system must be fast enough to at least accommodate current production speeds.

To assure robustness, several concepts are incorporated into the system framework design. 1) A combination of bottom-up (data-driven) control and top-down (model-driven) control is used to perform scene analysis. This combined or mixed control strategy is necessary to make the system robust and flexible. Advantages of the mixed control strategies will be discussed in detail in a later Section. 2) Hierarchical object models that consider defects at various levels of representation are used to recognize the defects. 3) A method for uncertain reasoning is incorporated into the vision system framework because there is uncertainty in all stages of visual processing. Local errors are introduced by the imaging and digitization process, by imperfect knowledge sources (e.g., segmentation) as the result of incorrect heuristics, or by inherent ambiguity in the scene itself.

To make the vision system flexible, the knowledge structure used is modular. It is important to keep a priori problem-dependent knowledge only for a particular web inspection problem separate from general knowledge for a variety of web inspection problems in order to make the system easily adaptable to different web inspection problems. It is also important to organize the problem-dependent knowledge in a modular manner so that it can be easily adapted to a change of operating environment, so that adding new knowledge or deleting obsolete knowledge can be easily accomplished. As will be described in detail later, the Blackboard framework provides this modularity of knowledge sources.

The third and fourth requirements force the vision system to handle and analyze a large amount of image data, at least at the speed of current human inspection. These requirements are addressed by attempting to create a framework that minimizes the computational complexity of the analysis task. To achieve this goal, several methods were used. 1) For segmenting images of material to be inspected, a histogram-based multi-thresholding is used because of its computational simplicity. 2) Other analysis algorithms are also optimized to minimize computational complexity. One way this optimization is accomplished is by using a focus of attention mechanism to guide the scene analysis process. This focus of attention mechanism marks candidate regions so that they will be operated on by the most appropriate defect detection procedures. This marking is accomplished using computationally simple methods. 3) Knowledge about defects is partitioned according to defect type. This allows concurrent parallel processing by running each defect detection procedure on a separate processor, thus speeding up the system operation.

II. Vision System Conceptualization

Usually, a computer vision system, especially a knowledge-based vision system, consists of two stages: low-level vision and high-level vision. The low-level vision tries to extract features characterizing input image data through image processing operations such as edge detection and region segmentation. This involves transforming a numerical matrix representation of an image to a symbolic representation that is composed of a set of spatially related image primitives, such as edges and regions. Also various features are extracted from the primitives. High-level vision seeks to attach a consistent interpretation to these primitives and construct a description of the scene. This scene analysis is carried out using a priori knowledge about the scene's domain. Interpretation of real world images is difficult because of uncertainties arising when formulating hypotheses based on noisy image data and imprecise models about what objects might appear in the scene. Furthermore, different hypotheses can even conflict. Consequently, a knowledge-based approach is commonly adopted for high-level vision.

The computer vision system for industrial web inspection can be conceptualized into two components : a low-level segmentation module and a high-level recognition module.

2.1 The Low-Level Segmentation Module

The purpose of the segmentation module in industrial web inspection is to reduce the huge amount of image data of the material to be inspected by extracting regions that might potentially contain a defect. An objective is to do this as fast as possible. Since in web inspection problems defect area is very small compared to the area of normal surface, computational complexity can be markedly reduced if defect regions can be extracted using simple methods at an early stage of the processing. Once this has been done, higher level processing that is typically more sophisticated and time-consuming can be applied only to those regions believed to potentially contain a defect. Certainly, no segmentation method can produce perfect boundaries of defects. However, it is only important for the segmentation process to detect a key part of each defect, even if the defect's accurate area and boundary cannot be detected. These key parts of defects can then be used as clues for invoking top-down processing methods in the high-level module to find a more accurate boundary for each such defect.

There exist two major approaches to image segmentation: edge-based and region-based [6]. For a survey of segmentation, refer to [6, 7]. In edge-based methods, edges or local discontinuities in gray level are detected first using an operator and then connected to form longer, hopefully complete, boundaries. There are two problems with edge-based segmentation methods. First, edge detection methods often produce a number of "false" edges due to image "noise" where no meaningful scene boundary exist. Secondly, there are actually locations on an object boundary that cannot be detected using only local information, but require the use of global contextual information. These problems may be somewhat alleviated by using recent advanced edge detection methods such as [8, 9]. However, in industrial inspection, images can be very large and applying an edge detector to every pixel can be time-consuming. Thus, edge detection methods should be selectively applied only to areas where edge information is needed for image analysis.

In the region-based segmentation, a connected set of pixels is found that share a common property (such as intensity or color) [6]. Methods used for the region-based segmentation can be divided into two categories: 1) methods that are directly applied to images (e.g., region growing [10] and split/merge methods [11]) and 2) methods that are applied to a feature space of images where features are extracted from images (e.g., thresholding [12]). Note that methods in the first category have a computational complexity that is proportional to the number of pixels in the input image. The computational complexity of methods in the second category depends on the number of features used and the complexity associated with feature extraction. Gray-level histogram-based thresholding is the most efficient of methods in the second category because it works on a 1-dimensional feature space (the gray-level histogram) and the features (gray-levels) are obtained directly from the original image. Also, this method is much more efficient than any method in the first category. This speed difference is particularly significant for high resolution images such as those typically encountered in web inspection problems.

The above edge/region based segmentation methods are based on the assumptions that objects consist of relatively homogeneous surfaces or that intensity changes correspond to object boundaries. In the texture images, these assumptions are no longer valid. To segment texture images, a texture operator is typically required. The size of the texture operator should be larger than that of the fundamental elements that comprise the textures. A number of texture segmentation methods have been reported in the literature. A survey of these methods is given in [13]. Texture operators are too computationally complex to be used to segment images for industrial web inspection. These operators also suffer from the fact that accurate boundaries of defects cannot be obtained because texture measures are usually calculated from large regions, not just a few pixels. Of course, some problems, e.g., inspection of sand paper, require use of texture segmentation methods. However, in general, it is advisable to apply texture operators selectively only to areas where texture measures are absolutely needed.

The above discussion of the various segmentation methods leads to the conclusion that a gray-level histogram based thresholding method seems to be the most appropriate segmentation method for use in computer vision systems for industrial web inspection. If texture features and edge features are extracted, they should be extracted at a later processing stage and should be computed from only those areas where these features seem to be necessary for more detailed analysis.

2.2 The High-Level Recognition Module

The purpose of the recognition module is to perform the scene analysis task. It analyzes each of the regions forwarded to it by the segmentation module and identifies the type of defect present in each of these regions. It will be argued that the vision system requirements force one to incorporate artificial intelligence methods into the design of the recognition module..

Conceptually, there are three basic approaches to scene analysis [14]. The first of these is the bottom-up type approach. Using a version of this type of approach image data is processed by a number of different operations, each operation producing a new data structure that makes some new facet of the image explicit. The last of the operations performed are those that actually label the region of an image. Bottom-up approaches are known to be very sensitive to noise. Any mistake made by an early processing operation propagates up through the rest of the processing operations. As such, this type of approach has proven ineffective on real world images [14].

A second class of scene analysis strategies is top-down methods. The basic idea behind a top-down method is the formulation of a hypothesis about what is in the image. Once the hypothesis has been made operators are applied to the image to verify whether the formulated hypothesis is correct. Note that the initial hypothesis is generated without using any information collected from the image. Further, if the results of an operation disprove the current working conjecture of the scene analysis system, then another working conjecture or hypothesis is generated. The generation of this new hypothesis is also independent of any information obtained from the image. Because no image derived information is used in formulating working hypotheses top-down methods are very limited in their generality. However, these approaches have been successfully used on very complicated albeit highly structured real world scenes, e.g., the analysis of chest radiographs [15].

The third class of scene analysis strategies is the combined, mixed or heterarchical strategies. Such strategies use a combination of both bottom-up and top-down methods. Image derived information is used to guide the analysis and to formulate hypotheses -- bottom-up processing. Special operators are then applied to the image to verify the hypotheses -- top-down processing. It is generally agreed that a combined strategy is the most robust type of control strategy. It is also generally agreed that human vision uses a combined strategy

Creating this or any other industrial inspection system requires that the software be easily modifiable, allowing easy incorporation of additional algorithms for improving accuracy or the removal of algorithms that are unnecessary in order to meet established accuracy criteria. One way, perhaps the only way, that this can be accomplished is to have the program be made up of a number of loosely coupled modules that communicate in an explicit manner.

On real world applications the need for robust vision systems typically requires a combined strategy. An important part of such combined strategies involves making hypotheses about what is present at a particular location in an image. It also involves testing these hypotheses by performing operations on the image and examining the results. Finally it involves keeping track of all competing hypotheses, ordering the attempts to verify each based on which hypothesis is believed to be the most probable. This implies the need for a belief maintenance system.

These arguments all point to the fact that a pattern directed inference system should be an important part of vision system design. In particular, a production system seems an appropriate choice for the system. A design based, in part, on a production type system would certainly provide a mechanism for providing the loosest possible coupling among program modules. The "if" part of each rule would specify the conditions under which a particular module should be executed. The execution of the module would be the "then" part of the rule.

Incrementally adding new program knowledge or removing unneeded program knowledge involves either adding or removing rules from the system. This is facilitated by the fact that the conditions for executing a module are explicitly given in the rule. Adding or removing rules does not require that other rules of the system be changed. It merely alters how the system responds to given sets of circumstances.

Ordering attempts to validate competing hypotheses is also easily accomplished. Each hypothesis is an assumption that a particular defect is present at a particular location in the image. Associated with each hypothesis are rules for validating it and "conclusively proving" that that defect is the one present. Structuring the knowledge base by grouping the validation rules by hypothesis represents a natural method for improving overall efficiency. This restricts the number of rules that must be considered in performing the match to see which rules might be applicable for firing at any given instant in time. Associated with each hypothesis is a belief. The hypothesis that is currently considered to be the most likely becomes the working hypothesis, the one whose rules will be fired by the system. A change of hypothesis causes a change in the rules that are considered.

With regard to object models for the various defects, robust vision methodologies typically demand hierarchical models to be used, ones that consider objects at various levels of abstraction. This implies that the knowledge base takes on much more structure than that described above.

Practically speaking it is important to use as few levels of description as possible since doing so can markedly affect not only the computational complexity associated with belief maintenance [16] but also the complexity of the structure that must be imposed on the knowledge base.

These arguments suggest that a production system represents a very good programming paradigm to use to address a number of the tasks this industrial inspection system must perform. Note that the tasks for which this paradigm seems most appropriate are all relatively high level tasks.

In this Section it is argued that a robust vision system for industrial web inspection system should have the following characteristics.

1. Segmentation should be based on some type of gray-level histogramming operations.
2. The segmentation may have to incorporate certain characteristics specific to the web inspection problem in order to obtain required segmentation accuracies.
3. The control strategy used in the high-level recognition module should be a combined strategy employing both bottom-up and top-down components.
4. Object models used to recognize defects should be hierarchical in nature.
5. A production system type paradigm should be used in the software implementation of the vision system.
6. A belief maintenance system is required to help the vision system cope with the processing results obtained from noisy image data, matching against imprecise object models, and choosing the most appropriate hypothesis from a series of competing hypotheses.

III. Computer Vision Systems : A Review

Most vision systems that are currently in use in industry were developed with one specific inspection task in mind. Hence, the resulting machine vision system is typically inflexible in the sense that it is not easily adapted to other applications.

Yoda et al. [17] described an inspection system that can reliably detect submicron defects on multilayered LSI wafer patterns. Hara et al. [18] developed a printed circuit board (PCB) pattern inspection system based on fluorescent light optics. These systems are very limited as to the number of applications to which it could be applied.

Only a very few industrial inspection systems have been created that are knowledge-based. One reason for this might be that real-time processing is usually a critical requirement in industrial applications. Unfortunately, by their very nature, knowledge-based vision systems can be somewhat computationally complex, and hence achieving a real-time implementation of such systems has been difficult in the past. However, the rapid evolution of computers will allow a cost effective real-time implementation of such systems.

Darwish and Jain [19] used a rule based approach for detecting defects in printed circuit boards. Their approach is composed of segmentation and rule-based verification of defects. Testing of different regions is required to ensure that they meet the design requirements such as minimum (maximum) pattern width and minimum spacing between patterns. These requirements are formalized by a set of rules. Some of the advantages of this system are that it is fairly modular, so it is flexible in the sense that models and design rules can be adapted to different specifications, and that it uses two levels of representations for patterns, segmented regions and segmented line drawing. On the other hand, it still lacks some desirable features needed in robust vision systems. 1) Its control is done in a bottom-up manner. 2) No uncertainty management approach is employed. 3) No focus attention mechanism is used. Hence each segmented region must be tested to see if it meets the design rules. This is very inefficient.

Sanz and Petkovic [20] presented a prototype system for automated inspection of thin-film disk heads. For defect classification, a rule-based system was used, whose inputs are both geometrical and gray-level features extracted from previously segmented regions. One of the desirable characteristics of this system is that the choice of a rule-based system for defect classification and comparison to inspection specifications allows flexibility in modifying, adding, or deleting defect classes and specifications. However, some disadvantages exist in this system. 1) Its control strategy is purely bottom-up. 2) Its rule-based classifier is not efficient because there is no focus of attention mechanism for determining which potential defect should be tested next. 3) No uncertainty management scheme is used. 4) There is only one level of defect descriptions, i.e., the features extracted from each segmented region.

Barlett et al. [21] explored two approaches to automatic solder joint inspection: statistical pattern recognition and expert systems. In the statistical pattern recognition approach, features are extracted from a subimage of the digitized PCB image containing a single solder joint. The expert system uses features more analogous to the visual clues that a human inspector would rely on for classification. Rules using these cues were developed and a voting scheme was implemented to incrementally accumulate classification evidence. Both methods compared favorably with human inspector performance. However, both approaches have some limitations. 1) Perfect registration of images is assumed. 2) Both lack desirable characteristics for a robust and adaptable vision systems such as combined control, focus of attention, hierarchical object models, and uncertainty management schemes.

More powerful and sophisticated knowledge-based vision systems have been developed for application areas where real-time processing is not required. These areas include aerial image interpretation [4, 22, 23, 24], outdoor scene interpretation [5, 25], medical image analysis [26, 27], etc. Notable systems are reviewed and compared with the desirable attributes described in the last section.

Nagao and Matsuyama [4] developed a vision system for automatically interpreting aerial images using structural analysis. The analysis process in this system is divided into the following steps: 1) image smoothing, 2) region segmentation, 3) extraction of characteristic regions, and 4) object-detection subsystems. The characteristic region is a set of regions that has characteristic properties, e.g., elongated, large, homogeneous. These characteristic regions are used to estimate approximate areas that are likely to contain objects. After extraction of characteristic regions, a set of object-detection subsystems performs knowledge-based analysis to locate objects in a scene. Each object-detection subsystem focuses its attention on specific local areas by combining several characteristic regions. Then it checks for the existence of specific objects by consulting the knowledge stored in the subsystem. All the information about the properties of elementary and characteristic regions and recognized objects is stored in a blackboard. All the subsystems communicate indirectly only via the blackboard. When a region happens to be recognized as multiple objects, the system cancels the recognition of other objects except the most reliable one. Nagao and Matsuyama's system has most of desirable characteristics described in the last section. One exception is that no uncertainty management scheme is used in this system.

The VISIONS system [28, 5], which operates on multi-spectral images of outdoor images, can be divided into two main parts: the low-level segmentation processes and the high-level interpretation processes. The segmentation processes yield an intermediate symbolic representation (e.g., regions and lines and their attributes) of the image data without making use of any knowledge about specific objects in the domain. At the beginning of the interpretation process, object hypothesis rules are applied to the region and line representation to rank-order candidate object hypotheses. This initial iconic-to-symbolic mapping provides an effective "focus of attention" mechanism for later processing. Reliable hypotheses are selected as image-specific exemplars, and extended to other regions and lines through an object-dependent similarity matching strategy. At this point, intermediate grouping is performed for merging and modifying region and line elements to match expected object structures. Verification strategies exploit particular spatial relationships between a hypothesized object and other specific expected object labels or image features. Feedback to the lower-level process for more detailed segmentation can be requested for correcting errors detected in the interpretation process. Scene knowledge is represented in a hierarchical schema structure organized as a semantic network. Each schema defines a highly structured collection of elements in a scene or object. A number of desirable features are embedded in this system.

IV. The Proposed Vision System

In Section II, it was argued that a production system type structure should be used as the knowledge system architecture of the proposed vision system. Among various kinds of production or rule-based systems, the Blackboard framework [29, 30] has been selected for this purpose. There are several advantages of the Blackboard framework over straightforward rule-based systems [30]. 1) Modularity: Inherent modular characteristic of the Blackboard framework allows easy design, testing, and maintenance of the system. 2) Dynamic control: This framework provides a wide range of capabilities for controlling the problem-solving behavior of the system including top-down, bottom-up, and combined control. 3) Efficiency: The Blackboard framework provides a convenient method for employing a focus of attention mechanism. Focusing the attention of the system allows the best data and the most promising methods to be exploited first. 4) Concurrency: The modularity and flexible control structure of the Blackboard framework can support concurrent parallel processing, which can play an important role in speeding up the system operation [5]. The Blackboard framework has been used in other very complicated problems such as speech recognition [29] and natural scene understanding [4, 5].

4.1 The Blackboard Framework

The blackboard framework is a problem solving model that is a highly structured, special case of opportunistic problem solving. It usually consists of three separate components: knowledge sources (KSs), blackboard data structure, and control modules [30].

The domain knowledge needed to solve a problem is partitioned into KSs, which are kept separate and independent. The condition component dictates when the KS can be used and the action component specifies the contribution that the KS can make. A KS can communicate with other KSs through only a global data base, a blackboard that records hypotheses generated by KSs.

The purpose of the blackboard is to hold computational and solution-state data needed by and produced by the KSs. The KSs use the blackboard data to interact with each other indirectly. A blackboard consists of objects from the solution space. These objects can be input data, partial solutions, alternatives, and final solutions. The objects in the blackboard are hierarchically organized into levels of analysis. Properties of objects on one level serve as input to set of KSs, which, in turn, place new information on the same or other levels.

There is a set of control modules that monitor the changes on the blackboard and decide what actions to take next. Various kinds of information are made globally available to the control modules. The information can be on the blackboard or kept separately. The control information is used by the control modules to determine the focus of attention. The focus of attention indicates the next thing to be processed. The focus of attention can be either the KSs or blackboard objects, or a combination of both. The solution is built one step at a time. Any type of reasoning step (data driven, goal driven, etc.) can be applied at each stage of solution formation. As a result, the sequence of KS invocation is dynamic and opportunistic rather than fixed and preprogrammed. However, in practice, it is more efficient to use both sequential and opportunistic scheduling of KSs, depending on the characteristics of KSs, rather than to use purely opportunistic scheduling.

4.2 A Blackboard Framework for the Proposed Vision System

Within a Blackboard framework, the proposed vision system can be conceptualized to have two modules: a low-level segmentation module and a high-level recognition module. The block diagram of the system is shown in Fig. 1.

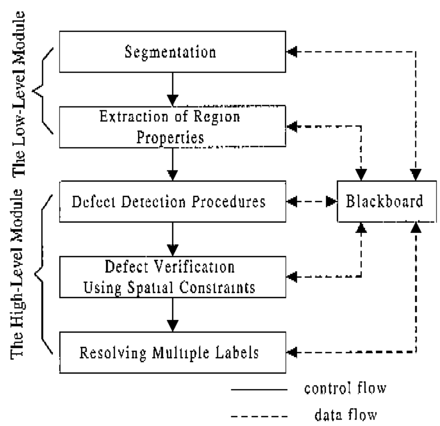


Fig. 1. Block diagram of the proposed vision system

The low-level module consists of two parts : segmentation of an input image containing the material to be inspected and extraction of region properties. The main purpose of the segmentation in the web inspection problems is to reduce the volume of the image data to be processed by extracting potential defects from the image. A multi-thresholding method is used to do the segmentation. (Refer to Section II for a detailed discussion about various segmentation methods.) The thresholds are adaptively and automatically selected on the basis of valley points and inflection points of the gray level histogram of the input image. A priori knowledge specific to web inspection problems is incorporated in selecting the thresholds in order to obtain segmentation results with acceptable accuracy. After segmentation, the low-level module identifies all connected regions, eliminates small noise regions, merges adjacent regions that have similar average gray levels, and computes region properties of merged regions. The result of this processing is a more accurate and concise description of each of the regions passed on for higher level processing.

The high-level module performs the scene analysis operation, i.e., recognizes defect areas. Basically, the high-level module

consists of three parts: a focus of attention mechanism, defect detection procedures, and verification using contextual information or spatial constraints. Confidence vectors are used as the focus of attention mechanism to screen candidate regions in an effort to determine the type or types of defect each region might represent. A region's confidence vector determines which defect detection procedures will be applied to that region.

Each defect detection procedure is designed to detect a particular type of defect. Each defect detection procedure operates completely independent of the other defect detection procedures. Each defect detection procedure is applied only to regions in the image whose confidence vector indicates it might be the corresponding defect type. In each defect detection procedure, top-down processing using special operators applied to the original image is used when more evidence is needed to support a hypothesis. Each defect detection procedure consists of two steps: initial labeling and labeling verification. The initial labeling step constructs the sets of connected regions, called DEFECT_OBJECTs. Each DEFECT_OBJECT is a set of connected regions all of which have been assigned the same defect label by a defect detection procedure. The label verification step tries to verify the assigned label of each DEFECT_OBJECT generated.

Finally, the label of each DEFECT_OBJECT is further verified using spatial constraints among adjacent DEFECT_OBJECTs. This is necessary because each defect detection procedure encodes knowledge specific only to its defect type and does not contain contextual knowledge among the different defect types.

Knowledge Sources

Several knowledge sources exist in the low-level vision module. The first one, SEG, performs a segmentation. A KS CCL does the connected component labeling. A KS SRE eliminates small noisy regions. A KS RMERGE (region merge) merges adjacent regions which are similar in average gray levels. Finally, a KS RPT extracts region properties for each region.

There are a number of knowledge sources for the high-level vision module. A KS, INIT, does some initialization. A KS CALCV calculates a confidence vector for each region based on its region properties, using fuzzy logic. The ith component of the confidence vector for region R is a confidence value that R is a defect of type i. LBLCON is a KS for labeling unlabeled regions using spatial contextual dependency. VRFDFC is a KS for verifying labels assigned to regions by defect detection KSs using spatial constraints between different defects. RMLBL is a KS that resolves labeling ambiguity if multiple defect labels are assigned to a region. Other KSs are defect detection procedures, each of which is designed to detect one kind of defect in the surface of the material being inspected. It is natural to partition knowledge about defects, according to defect type. The major advantage in partitioning knowledge this way is the ease with which additional defects can be incorporated into the system by simply adding new defect detection procedures.

Blackboard Data Structure

The blackboard structure is organized into the following entities:

Original image
Symbolic segmented image
Region property table
DEFECT_OBJECT property table
Region adjacency matrix
DEFECT_OBJECT adjacency matrix

The symbolic segmented image is a symbolic image where each pixel in a region is labeled with the same unique region number. The symbolic image is generated by the segmentation step in the low-level module. The region property table is a collection of property lists for all the segmented regions. A property list is associated with each region; it consists of a region attribute vector, a confidence vector, a pointer vector, and an interpretation label. The region attribute vector consists of various region properties whose values are assigned by the low-level module. The interpretation label is the label (normal surface or one of the defect types) assigned to the region. Confidence vectors, pointer vectors, and interpretation labels are assigned values by the high-level module.

There are two basic units in the blackboard: regions and DEFECT_OBJECTs, that are hierarchically linked by a part-whole relationship. A DEFECT_OBJECT consists of a set of connected regions that have the same defect label. The concept of the DEFECT_OBJECT is illustrated in Fig. 2. The ith component of a region's pointer vector points to a DEFECT_OBJECT with ith defect label associated with the region. For example, in Figure 2, if DO1 has ith defect label, the ith component of each pointer vector associated with regions R1, R2, R3, and R3 point to DO1. Note that one can determine whether a region is assigned multiple defect labels by simply checking how many DEFECT_OBJECTs are pointed to by the region's pointer vector. Each DEFECT_OBJECT generated stores its properties on the DEFECT_OBJECT property table. The DEFECT_OBJECT property table has the same structure as the region property table except for the pointer vector field. Each component of the pointer vector associated with a DEFECT_OBJECT is always 0 because there is no parent node of the DEFECT_OBJECT. If a DEFECT_OBJECT is generated and verified by a defect detection procedure that was designed to detect defect type i, the interpretation label of the DEFECT_OBJECT is assigned defect type i . The region adjacency matrix is defined as a matrix whose (i,j)-th element is the common perimeter between region i and region j divided by the perimeter of region i. The DEFECT_OBJECT adjacency matrix is defined similarly. These adjacency matrices provide contextual information among regions or DEFECT_OBJECTs that is vitally important in interpreting images.

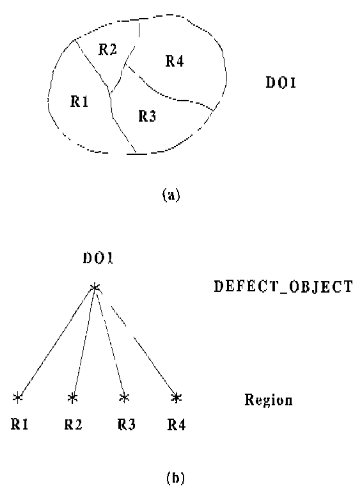


Fig. 2. A DEFECT_OBJECT consisting of four regions. (a) Four connected regions labeled as the same defect type. (b) A DEFECT_OBJECT DO1 generated.

Control

The control of processing flow is done using a combination of both sequential control and opportunistic control. This combination is more efficient than purely opportunistic control used in typical blackboard systems. Control in the low-level module is sequential. KSs in the low-level module are applied in the following order: SEG, CCL, SRE, and RMERGE. At an initial stage of high-level processing, KS INIT is applied for initialization that includes calculation of confidence vectors. Then, KSs for defect detection are run independently and opportunistically. A region's confidence vector is used as a "focus of attention" mechanism to selecting which defect detection KSs should be activated to process the region. Once KSs for defect detection have finished their operations, KS VRFDFC (verification of defects using spatial constraints) and KS RMLBL (resolving labeling ambiguity) are performed in sequence. Before and after applying VRFDFC, a determination is made as to whether there are any regions that are still unlabeled. If there are unlabeled regions, then KS LBLCON is invoked to label those regions using spatial contextual dependency.

V. An Application

On the basis of the proposed vision system framework, a computer vision system for automated lumber grading has been developed. The purpose of the vision system for automated lumber grading is to locate and identify grading defects on rough hardwood lumber in a species independent manner. This problem seems to represent one of the more difficult and complex web inspection problems. No vision system has been developed to tackle this problem. The main reason for this seems to be that there are a lot of factors causing normal surface discolorations, discolorations that must be differentiated from areas that truly contain a defect. Also, some defect types are difficult to recognize due to irregular characteristics of their shapes. Hence if one can use the previously described framework to develop a vision system for grading rough lumber, this framework should be versatile enough to work on a variety of other web inspection problems. In fact, the methodologies used in the vision system for grading lumber would seem applicable to a number of other web inspection problems as well.

The system has been tested on approximately 100 boards from several different species. Parameters that are not adaptively determined during the operation were fixed throughout all the experiments. The vision system has been found to be robust.

VI. Conclusion

In this paper, a general vision system framework has been developed that can be easily adapted to a variety of industrial web inspection problems. The objective is to automatically locate and identify "defects" on the surface of the material being inspected. This framework has the following desirable characteristics:

The vision system framework incorporates a number of methods that should help assure robust operation.

The vision system framework is flexible, so that it can be easily adapted to a number of different web inspection problems. Adapting the system to a particular problem domain only requires that new defect detection procedures be created and that new spatial constraints be formulated.

The design approach used in creating the framework attempts to minimize the computational complexity of the analysis so that real-time operation at industrial speeds will be possible.

References

[1] 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.

[2] R.T. Chin, SURVEY: automated visual inspection: 1981 to 1987, *Computer Vision, Graphics, and Image Processing*, vol. 41, pp. 346-381, 1988.

[3] R.T. Chin, "Algorithms and techniques for automated visual inspection", in *Handbook of Pattern Recognition*, T.Y. Young and K.S. Fu, Eds. Academic Press, 1986, pp. 587-612.

[4] M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*, Plenum, 1980.

[5] A.R. Hanson and E.M. Riseman, "The VISIONS image understanding system", in *Advances in Computer Vision*, Vol. 1, C. Brown, Ed. NJ: Lawrence Erlbaum Associates, 1988, pp. 1-114.

[6] R. Nevatia, "Image Segmentation", in *Handbook of Pattern Recognition*, T.Y. Young and K.S. Fu, Eds. Academic Press, 1986, pp. 215-245.

[7] R.M. Haralick and L.G. Shapiro, "SURVEY Image segmentation techniques", *CVGIP 29*, pp.100-132, 1985.

[8] D. Marr and E. Hildreth, "Theory of edge detection", *Proc. Royal Society London B*, vol. 207, pp. 187-217, 1980.

[9] J. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. 8, no. 6, pp. 679-698, 1986.

[10] C.R. Brice and C.L. Fennema, "Scene analysis using regions", *Artificial Intelligence*, vol. 1, 205-226, 1970.

[11] T. Pavlidis, *Structural Pattern Recognition*, New York : Springer-Verlag, 1977.

[12] P.K. Sahoo, S. Soltani, and A.K.C. Wong, "SURVEY : A survey of thresholding techniques," *CVGIP 41*, pp. 233-260, 1988.

[13] R.M. Haralick, "Statistical image texture analysis," in *Handbook of Pattern Recognition*, T.Y. Young and K.S. Fu, Eds. Academic Press, 1986, pp. 247-279.

[14] R.W. Conners, "The need for a quantitative model of human preattentive vision," *Proc. SPIE*, vol. 786, Applications of Artificial Intelligence, pp. 211-220, 1987.

[15] P.P Tsiang, "Computer analysis of chest radiographs using size and shape descriptors," *Ph.D. Dissertation, Univ. of* Missouri-Columbia, 1974.

[16] J. Barnett, "Computational methods for a mathematical theory of evidence," *Proc. International Joint Conference on Artificial Intelligence*, pp. 868-875, 1981.

[17] H. Yoda, Y. Ohuchi, Y. Taniguchi, and M. Ejiri, "An automatic wafer inspection system using pipelined image processing techniques," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. 10, no. 1, pp. 4-16, 1988.

[18] Y. Hara, H. Doi, K. Karasaki, and T. Iida, "A system for PCB automated inspection using fluorescent light," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. 10, no. 1, pp. 69-78, 1988.

[19] A.M. Darwish and A.K. Jain, "Rule based approach for visual pattern inspection," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. PAMI-10, no .1, pp. 56-68, 1988.

[20] J.L.C. Sanz and D. Petkovic, "Machine vision algorithms for automated inspection of thin-film disk heads," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. 10, no. 6, pp. 830-848, 1988.

[21] S.L. Bartlett, P.J. Besl, C.L. Cole, R. Jain, D. Mukherjee, and K.D. Skifstad, "Automatic solder joint inspection," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. 10, no. 1, pp. 31-43, 1988.

[22] D.M. Mckeown, JR., W.A. Harvey, JR., and J. Mcdermott, "Rule-Based Interpretation of Aerial Images," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. PAMI-7, no. 5, pp. 570-585, 1985.

[23] V. Hwang, L.S. Davis, and T. Matsuyama, "Hypothesis interpretation in image understanding systems," *Computer Vision, Graphics, and Image Processing*, vol. 36, pp. 321-371, 1986.

[24] D. Kuan, H. Shariat, K. Dutta, and P. Ransil, "A constraint-based system for interpretation of aerial imagery," *Proc. IEEE CVPR 88*, pp. 601-609, 1988.

[25] Z-N. Li and L. Uhr, "Pyramid vision using key features to integrate image-driven bottom-up and model-driven top-down processes," *IEEE Trans. Systems, Man, and Cyber.*, vol. SMC-17, no.2, pp. 250-263, 1987.

[26] H. Niemann, H. Bunke, I. Hofmann,G. Sagerer, F. Wolf, and H.Feistel, "A knowledge based system for analysis of gated blood pool studies," *IEEE Trans. Pattern Analy.*

and Mach. Intelli., vol. 7, no. 3, pp. 246-258, 1985.

[27] S.A. Stansfield, "ANGY: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. PAMI-8, no. 2, pp. 188-199, 1986.

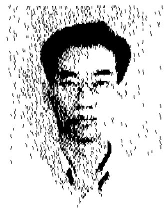
[28] B.A. Draper, R.T. Collins, J. Brolio, A.R. Hanson, and E.M. Riseman, "Issues in the development of a blackboard-based schema system for image understanding," in *Blackboard Systems*, R. Engelmore and T. Morgan, Eds. MA: Addison-Wesley, 1988, pp. 133-146.

[29] L.D. Erman, F. Hayes-Roth, V.R. Lesser and D.R. Reddy, "The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty," *ACM Computing Surveys*, vol. 12, no. 2, pp. 213-253, 1980.

[30] R. Engelmore and T. Morgan, Eds., *Blackboard Systems*, MA: Addison-Wesley, 1988.

Tai-Hoon Cho

He received the B.S. degree in electronics engineering from Seoul National University in 1981, and the M.S. degree in electrical and electronic engineering from KAIST(Korea Advanced Institute of Science and Technology) in 1983. In 1991, he received the Ph.D. degree in electrical engineering from Virginia Polytechnic Institute & State University. From 1992 to 1997, he was a Senior Researcher at LG Industrial Systems R&D Center. In 1998, he joined the School of Information Technology, Korea University of Technology and Education, Chonan, Korea, where he is currently an assistant professor. His research interests include computer vision, image analysis, and neural networks.



Young-Kee Jung

He received the B.S. degree in electronic engineering from Seoul National University, Seoul, Korea, in 1986, and the M.S. degree in electric & eletronic engineering from Korea Institute Science and Technology (KAIST), Taejon, Korea, in 1994. He is currently working toward the Ph.D. degree in Information and Communications Department of communications engineering from Kwangju Institute Science and Technology (K-JIST), Kwangju, Korea. From 1986 to 1999, he was with LG Industrial System Laboratory, Anyang, Korea, where he was involved in development of machine vision and traffic surveillance system. Since 1995, he has been with Honam University, where he is currently Professor of Computer Engineering Department. His research interests include image processing, object tracking and visual surveillance.

Hyun-Chan Cho

He received the B.S. degree from the Department of Electronics Engineering in 1983 from Kwangwoon University and M.S. and Ph.D. degrees in Department Electronics Engineering from Chung-Ang University, in 1985 and 1991, respectively. He is currently an associate professor at the School of Information Technology, Korea University of Technology and Education(KUT), Chonan, Korea. His research interests include fuzzy logic control, neural networks, genetic algorithm and its applications. He was a visiting scientist in the Department of System Science and Industrial Engineering of SUNY, Binghamton during the years 1996-97.