MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

Recent Progress in Industrial Machine Vision

Omid Mohtadi
ESLAI, Argentina
IBM Argentina

Jorge L. C. Sanz
IBM Almaden Research Center, USA
IBM Argentina

Abstract

In this paper, a survey of recent progress made in machine vision systems is presented. The main focus of the paper is on industrial inspection problems. Several important contributions recently reported in automated visual inspection of integrated circuits, printed circuit boards, packaging, disk-heads, metal surfaces, and other products are revisited.

1 Introduction

Machine vision methods and technology applied to industrial inspection problems have received a great deal of attention in the last decade, [9]. A number of new systems for a variety of applications have been reported in the literature in the last two years. This paper attempts to survey some of the most relevant applications and solving approaches recently documented in the literature. This survey is by no means exhaustive, but it does help identify new contributions and provides details of proposed methodologies.

The sections of the paper are organized according to applications. In each section, different contributions involving systems and related machine vision methodologies are surveyed. The depth of the material is intended to provide a self-contained description of the algorithms and systems, although for further details, readers are referred to the technical report format of this paper [28] and to the original contributions. It is hoped that presentation of the material in the paper will help identify common machine vision methods arising from different systems.

2 Printed Circuit Boards

2.1 Printed Circuit Board Inspection Using Fluorescent Light

PCB defects such as shorts, cuts, mousebites, surface separation are detected by using violet or ultraviolet illumination and detecting the patterns using a high sensitivity TV camera. This sensing captures the emitted flourescent light (FL) by the base material after filtering.

A working system using the above sensing ideas was developed in [19] which uses emitted FL and reflected light together to be able to detect and classify all possible defects. Basically the system works by subtracting two grabed images, taken from two PCBs, so that differences between these two images reveal defects. The comparison algorithm is "intelligent" in the sense that it selects the boundaries and narrow sections of the pattern and compares them. see details in [17, 18, 16]. Even in the presence of deformations in the patterns or slight registration errors, line shaped defects can be detected. The PCBs are placed on an X-Y stage and are scanned in the X and Y directions. The system also includes defect recognition and control sections. Before inspection accurate registration is done. But even when the process has begun, deviations from perfect registration is detected and corrections are made automatically. This allows the correction for slight expansions or contractions of PCBs. When a defect is detected its position coordinates are stored in a disk and the subimage containing the defect is displayed.

There is a defect usually referred to as surface separation. This type of defect is generally not detected by the emitted light sensing approach. Hence the proposed technique incorporates also reflected light-based methods which can detect this kind of defect.

The image processing is performed by pipelined hard-wired logic circuits. Image features are extracted from temporary local image memories by way of wired logic operators. Comparison of a pair of feature extracted patterns are also performed by logic circuits wired to local image memories.

The system is able to detect a defect of width 0.01 mm. The time consumed in the inspection of a 500mm $\times$ 600mm PCB is 18 minutes approximately. 100% defects were detected (compared by electric testing results). False alarms: less that five occurances per 500mm $\times$ 600mm PCB. Defects resulting from photomasks can not be detected. This is because this system compares patterns on actual circuit boards. To detect defects resulting from photomask defects, it would be necesary to compare patterns made from different photomasks.

According to the statistics given in [19], the system seems to have an exceptional accuracy, suitable for high-density PCBs (min. pattern width 0.08 mm, with pixel element size is 0.01 mm).

2.2 Solder Joint Inspection

A good solder joint detection method must be able to classify the defects not just make pass/no pass decisions. Traditionally two inspection methods are used to achieve this: human operator, and electrical testing.

Nakagawa in [29, 31, 30] exploited the deformation of an intense light beam incident on the solder joints, to acquire 3-D structural information of the joint. The algorithm is fast, is not influenced by high gloss of the solder joints, and is relatively insensitive to ambiental illumination conditions; but its speed is limited by the mechanical positioning of the beam. Accept/reject decisions are accurate but defect classification is not quite satisfactory. Slight positioning errors of the light beam can produce large classification errors.

McIntosh [27] uses special scene illumination techniques and camera filtering to accentuate the joints and to suppress the background. This allows for the extraction of binary image features used in the inspection algorithm. The system has low cost, and according to the author it has 99.8% flaw detection rate at 120 ms per solder joint. But it seems that binary images are not rich enough for a detailed defect classification.

Currently, there are a number of commercially available non-visual solder-joint inspection systems: Vanzetti [44]: the author uses infrared signatures of joints after heating them with a laser beam. This method also has limited speed due to the mechanical positioning of the beam, positioning errors can produce large classification errors, and heating the joints by the laser is questionable.

In [21, 22], a method based on radiographic imaging was presented. The system exploits shape and volume properties of the X-ray images for classification. This method has shown to be very accurate and is particularly suitable for Surface Mounted Devices (SMD) with hidden leads. The detection rate is reported to be in the range 95%-99% with 5% false rejects.

In [4], the problem of automating the visual inspection of pin-in-hole solder joints is considered. Two approaches are presented: statistical pattern recognition and expert systems. The Objective Dimensionality Reduction (ODR) algorithm is presented to enhance the performance of the statistical method.

Statistical pattern recognition approach

This approach consists of adequate feature selection and classifier design. Twenty seven features were selected by the authors by studying the distribution of their values for each class. Details of their computation can be found in [5]. These features can be divided into 5 categories:

- Basic gray-level statistics features: for each subimage containing a solder joint, 4 general features are calculated as follows: Normalized mean gray-level, the minimum normalized mean gray-value, and the percentage of dark and bright pixels. Also 2 application-dependent gray-level features are calculated: the normalized gray-level volumes of the central and outer frame subwindows.
- 3-D gray-level inertia features: four 3-D gray-level inertia features are used. These features estimate the first principal moment of inertia of the subimages, the sum of all three moments of inertia (x, y, and intensity), and the ratio of the brightness moment to the average of the two spatial moments of inertia.
- Faceted gray-level surface area features: 2 features are used. These compute the faceted gray level surface area, and the surface area obtained by summing the gray-level surface metric determinant over all pixels. These features are illumination independent.
- Differential geometric gray-level surface curvature features: 9 features are used. These include the average value and the percentage of the positive and negative Gaussian curvature pixels, average values and percentage of the positive and negative mean curvature pixels, and the quadratic variation of the gray-level surface. These features are also illumination independent.
- Binary image connected-region features: 6 features are computed for each thresholded subimage. These include the number of four-connected regions, the number of pixels in the largest four-connected regions, the number of pixels in the thresholded image that are not in the largest region, the ratio of the area of the min/max box around the largest region, the aspect ratio (width/height) of the min/max box surrounding the largest region, and the ratio of the perimeter squared to the area for the largest region within the subimage.

The authors in [4] used a minimum-distance classification algorithm. In this algorithm no assumptions are made about the probability density function of the data, and simple distance metrics are used for classification. Generally this kind of algorithm consists of two steps:

- Training: a set of solder joints that best represent the class that they belong to is chosen. Then average features are calculated for each class.
- Classification: given the feature vector of a solder joint, a distance metric of this vector with respect to the average feature vector of each class obtained in the training phase is evaluated (the norm can be Euclidian norm, for example). The solder joint is considered to belong to a class with minimum distance to the feature vector of the solder joint. It is obvious that in order to get acceptable performance a large amount of samples of each class must be available during training stage. This is not always possible. On the other hand, this simple algorithm is unable to decorrelate feature data, also it can not identify features that contributed insufficient information. The accuracy of the algorithm is further degraded when there is some clustering of classes for some features, which was the case in the solder joint inspection. The poor performance of the algorithm (i.e. good/bad decision 91.8%, correct classification 70.5%) testifies the above observations.

To overcome the above limitations ODR was designed (See the appendix in [4]). ODR is a multiple class multiple feature technique that decorrelates feature data and automatically weights features according to their contribution to the decision making process. It has been shown that the ODR, involving in this problem 9 classes 27 features, runs 3.45 times faster than straightforward Maximum Likelihood Multi-Variate Gaussians (MLMVG). Its detection rate is about 97.5% for pass/no pass decisions, and 91.3% for correct classification.

An expert system for solder joint inspection was also designed and compared to statistical classifiers. The average detection rates are the following: correct pass/no pass decisions: 96%, correct classification: 86%. It is obvious that the expert system performed better than the minimum distance classifier, but as was stated above ODR performed even better (see [4] for details).

Implementation

Intense, diffuse, uniform fluorescent light can be used for illumination purposes. A standard CCD and associated hardware, and a general purpose computer are suitable to capture and process the images.

In situations where great amount of training samples are available, accuracy is of paramount importance, expert knowledge is not accessible, or interaction with people is not required, statistical systems are the best choice. Otherwise, expert systems are more promising, specially when the system has to mimic the reasoning of human operators or when good heuristics for classification exist. However, a more sophisticated man/machine interface must be designed for expert systems so that rule-writing becomes easier. Also for industrial implementation, an interface between the PCB design data base and the inspection software would have to be built so that the inspection system would know what kind of joints are present and where they are located.

2.3 Design Data-Based Inspection for Printed Circuits

In [40], an approach for PWB inspection using CAD data as a reference is presented. Unlike other systems based on pixelwise comparison of the object under inspection with a "perfect" product, this system uses high level reference data. These CAD files are converted into descriptions of the pattern borders. Each border description is assigned a tolerance zone that determines the deviations allowed for patterns to be inspected. Such high-level CAD information is much more compact than pixel-level information. One of the objectives of the system is defect classification and not just accept/reject decisions. Following these ideas an experimental system was built and tested.

The system

The inspection methodology is the following. First, the edges of the printed circuit are detected and approximated by line segments. After alignment error correction, the line segments are compared to the reference data and the differences found are delivered to an error-analysis module for defect classification. Both the defect detection and analysis stages uses the same reference data base.

Photoplotter files are used to generate reference data. These files describe the wiring patterns by aperture definitions , exposure, and movement commands. However the photoplotter files do not contain all the information needed for inspection because for example hole descriptions and tolerance data are missing. For simplicity, tolerance zones assigned were proportional to the dimensions of the patterns. The primitives of the reference data generated from the photoplotter file are arcs and straight lines that describe the edges of the patterns.

The image capturing mechanism is rather conventional: CCD line scan camera, x-y table, and an image resolution of 10 or 25 mm depending on the camera lense used. Since the captured image is processed with a fixed threshold before further processing, special attention is paid to the illumination arrangement. In this case, one has to detect patterns with reflowed solder coats which show 3-D structures. For such materials, a combination of an integrating sphere and beam splitter was developed.

The image acquisition speed for a PWBs is 5Mpixels/s at $10/mum$ resolution and 7 mega-pixels/s for back-lighted mask films at the same resolution.

First, the edges of the patterns in the binary image are detected. Then a kind of approximation algorithm is applied to convert pixel-level data of the edges to higher level approximating line segments. The approximating line segment is started at a pixel and in successive scans more pixels are incorporated. The line ends (and a new one begins) when segment's orientation departs from 0, 45, 90, or 135 degrees that are principal line angles in PWBs. This line approximation is done in the raster-scan mode.

In order to be able to compare the line segment approximation against the CAD-based reference data, corrections of possible orientation errors in the PCB must be compensated for. The allignment method used is the one presented in [38].

Two main approaches are followed for defect analysis. In the statistical approach, a set of numerical features are computed for each defect candidate and the feature vectors obtained are classified. The experiments with the statistical approach were initially based on six features of the defect candidate (closed/open chain of line segments, inside/outside of conductor, parallel/not parallel with the model edge, number of model edges connected to, maximum distance to the nearest model edge, and minimum distance to the next nearest model edge).

The defect classes used were shorts, opens, too wide and too narrow conductors, spurious copper, and holes. The classification accuracy for test material consisting of over 250 simulated defects on an artwork (i.e. photomasks) was over 85%. On the other hand, for PCBs additional features were needed (about 20 features in total) to obtain 80% detection accuracy.

Structural analysis was found to be more useful in many cases. In this method, the structural relations between the defect candidate and the reference edge descriptions are determined. A defect is classified by finding the best match for it from among the defect models. Matching is performed by means of a depth-first search similar to the one in [14]. The state space searching starts with two sets of primitives, one for the defect candidate and the other for the model. When the search is terminated the matches are described by paths from the initial to the final states. Each state carries a weight that depends on the similarity between the associated primitives. The weight of the path is normalized by the number of nodes in the model.

The defect candidate graph is compared with each model and is assigned to the class for which the minimum normalized weight is obtained.

Experimental results and conclusions

The detection rate is about 85% for artwork and 80% for PWBs. The false alarm rate is high as the system generally classifies dust particles as spurious copper. Also, minimum detectable defect size is not specified. To this connection, line approximations of edges of the patterns introduces early approximation errors and obviously limits the minimum detectable defect size.

3 Integrated Circuits

There are two principal requirements imposed upon an industrial system for multilayer wafer inspection, namely, high speed and low false-alarm rate. Due to high density of recent semiconductor devices, defect sizes tend to become smaller and smaller. Therefore, the quantity of the data to be processed makes some kind of pipeline image processor indispensable. On the other hand, several acceptable variations from the ideal model must be taken into account if low false-alarm rate is desired. Some of these harmless variations are due to the following:

- Random texture on wafer surface caused by shading due to uneven grains.
- Small changes in brightness and pattern widths in each layer.
- Relative positional differences of patterns in each layer.

3.1 Inspection of Memory Wafers

A system for inspection of multilayered wafers is developed in [45]. The inspection speed achieved is about 6 minutes/cm^2. The heart of the inspection algorithm is a self reference technique in which two adjacent cells are compared and differences are considered as defect candidates. This methodology exploits the repetitive nature of cells in memory wafers. Also, some morphological operations are applied on this candidate "defect image" to suppress false alarms due to noise and tolerable pattern variations. Most of the steps are governed by CAD data describing the integrated circuit layers. All the image processing steps are performed in a one-pass fashion in a high speed pipeline image processor.

Sensing

The image is captured using a CCD linear array sensor and converted to 8-bit gray levels at 7 MHz video clock rate. The signal is continuously fed into a Real-Time Correction Circuit (RTCC) and Delay Circuit (DC). The RTCC which is initialized by an estimate of the repetition period, measures the actual period as a deviation from the initial value. Hence RTCC can adjust the delay time of the One Period Delay Circuit (OPDC), and image signal from the current cell A(t) can be precisely registered with the signal from the previous cell, B(t). The result of the comparison of A(t) and B(t) is the binary defect image which is fed to defect classification module which calculates some features of the defects and list the most relevant ones and stores them in the result memory.

The function of the calibration circuitry is to calculate a histogram and a projected distribution for a fixed area of the image which are used by the control computer to calculate parameters such as different threshold values, and coefficients of the equation used for mapping the design pattern data to actual wafer pattern coordinates. The Design Pattern Generator (DPG) stores bit-map data of repetitive design pattern and structure in each layer in advance. These bit-map data are aligned with each other using the actual displacement data between layers measured by the calibration circuit.

The inspection algorithm

If A denotes the current image signal, and B its delayed periodic counterpart, the absolute value of the difference between A and B is thresholded and a binary candidate defect image is generated. Two threshold values are used. The lower threshold produces binary images (G1) of larger blobs corresponding to true defective pixels and false alarms while the higher threshold generates smaller blobs (G2) representing the nuclear regions of the defects.

G2 is broadened within the G1 image. The resultant image is further submitted to a series of morphological operations to eliminate noise and unify defects. All these operations are done in real-time with a 2-D local pattern extraction circuit combined with logical AND and OR operations for erosion and dilation, respectively. The threshold values and the number of iterations of the morphological operations is entirely controlled using bit-map data of the design patterns. In this sense, this procedure is "adaptive", since the parameters can vary according to the textures in each layer and some other relevant information. The resultant image, called "defect image", is forwarded to the defect classification module.

The function of this module is to measure some properties of the defects to identify only the fatal defects. In this module, the area of the defects, and projected lengths on the X and Y axis are used as features. To facilitate the task of feature calculation in raster fashion, some sort of shape filtering is used to eliminate troublesome patterns, this filtering operation results in a defect image with no small holes or downward bays. Subsequently, a kind of region analysis is performed to evaluate these features. The feature measurement circuits output these feature values together with their center positions. These are then compared by decision logic circuit with corresponding threshold values specified for each pattern. The location of the defect within the design pattern is found and used for classification by using simple rules.

Inspection results and conclusions

The system is currently used for production control of memory devices such as 1Mbit DRAM and 256 Kbit SRAM in a 1.3μm CMOS process. The resolution is 0.2μm per pixel. The minimum detectable defect is in the order of 0.6μm, with an acceptable false alarm rate. The inspection speed is reported to be about 6 minutes/cm^2. It is estimated that more than 95 % of defects are detected. For practical purposes a small portion of the wafer is inspected exhaustively and a statistical decision is made.

The system has a high defect detection rate and very low false-alarm rate. The speed compares favorably with other systems such as P300 (about an order of magnitud faster).

3.2 Scanning Electron Microscope-Based Stereo Analysis

In [23], a novel technique for analyzing stereo images generated from SEM is presented. The new method uses binary linear programming approach to set up and solve the correspondence problem. It also uses constraints based on the knowledge of SEM image formation. The result of the application of the method on real images of IC's is also given in [23].

There are several parameters of the etched surface that are crucial to the controlling of the lithographic process, such as height of the step, slope of the side wall, and presence of undesired material. The authors in [23] discuss a novel method for the measurement of these parameters. Some basic requirements of a semi-conductor wafer inspection system is that the method must be non-destructive and that the method must have sub-micron accuracy, considering these requirements SEM stereo seems potentially adequate.

SEM-based stereo sensing is quite simple, the stereo image pair can easily be obtained by tilting the specimen and taking its images at two different angles. However, the principal computational cost relies on the solving of the correspondence problem. This is done mostly interactively. Once the correspondence is established 3-D depth information can be obtained using a set of 3-D reconstruction equations, derived from the geometry of the image formation.

The technique proposed in [23] is based on matching contours of relevant points in the pair of stereo images. The contour matching problem is posed as an optimization problem. Given two disjoint sets of contours, the contours matching problem consists of finding the "best match" between pairs of contours. The requirement is that each contour must have only one match, and the "best" solution is defined as a match resulting in the minimum difference in the lengths of the matched contours.

The contour matching problems that are discussed in [23] are posed as binary linear programming problems that have the special form: max $p.x^t$ over all binary vectors x subject to $C.x^t \leq b^t$. Where p is the objective function coefficient vector, x is the vector of variables with binary elements, C is the constraint coefficient matrix with binary elements and b is the constraint right hand side vector.

The most popular techniques for solving the binary programming problems are based on implicit enumeration via branch and bound techniques. The variant used by the authors is the one presented in [20].

SEM stereo correspondence problem

Solving the stereo correspondence problem requires two steps: the detection of predefined features in each image, and matching of these features. The predefined features are called *matching primitives*.

Choice of matching primitives

One has to do a careful selection of matching primitives since shape from stereo algorithms can return the depths of only those 3-D object points which correspond to feature points in both images. Some factors affecting the selection of features include the following:

- Feature points corresponding to high surface curvature points on the sample should be used. The idea is that the surface interpolation/approximation step that follows feature-based stereo matching will use this points as control points to fit smooth, low-order polynomial patches.

- The likelihood of the presence of the match of a feature point in the other image should be high.

- It should be possible to spatially locate corresponding feature points in the images accurately.

The authors used as primitives points where the curvature of the image intensity function achieves a local maximum with a sufficiently large magnitude, (referred to as high image curvature points) along with intensity edges.

Constraints for the matching problem

These constraints are derived from the imaging geometry, the physics of the image formation, and the geometry of the 3-D scene being viewed. The proposed stereo matching constraints are the following:

- The epipolar constraint: this constraint states that the match of an image point on one epipolar line must lie somewhere along the corresponding epipolar line in the other image.

- The uniqueness constraint: this constraint states that for each feature point in one image there can be at most one matching feature point in the other image.

- The ordering constraint: this constraint states that the ordering of feature points along an epipolar line in one image should be the same as the ordering of the corresponding match points along the epipolar line in the other image.

- The depth/height boundedness constraint: this constraint states that the computed depth/height (or disparity) values should be bounded. It is based on the assumption that objects in the scene have finite depth. This constraint is enforced by limiting the search for the match of a point in the other image to only a section of the corresponding epipolar line.

- The surface smoothness and figural continuity constraint: the surface smoothness constraint states that the computed 3-D surface points should not depict a surface which has abrupt jumps. The figural continuity constraint enforces smoothness of disparity along feature contours in the image.

The algorithm

It is now considered how the stereo matching problem defined by the above specifications can be transformed into an equivalent binary programming problem.

Each possible match of compatible left and right image contours is represented by a binary variable which indicates the acceptance/rejection of a match respectively. The number of points that are matched is considered as the cost associated with that match. The objective function that is maximized is $\sum p_i . x_i$ where p_i is the cost associated with accepting the contour match represented by x_i. p_i can be chosen to count the number of effectively matching points on the two contours, along with other measures based on feature values that strengthen the contour correspondence. The constraints are in the form of inequalities where the variables have a coefficient of 1 and the right hand side is a 1 or 2 (i.e. they look like $x_i + x_j + \ldots \leq 1$ or 2).

The steps of the algorithm are the following:

- Feature detection: each epipolar line in each image is convolved with one-dimensional Gaussian filters having s = 0.4 and 6.4. The DOG profiles are obtained. High curvature points and/or zero-crossing points are detected on each DOG profile in both images.

- Contour detection: a connected component algorithm is used to get high curvature points and zero-crossings feature contours.

- Variable and cost of variable identification: based on the "similarity of feature points" and the "boundedness of disparity" constraints, possible matches between feature contours and their associated costs are identified. A binary variable is assigned to each such contour match.

- Constraint generation: each of the constraint generators independently generate constraints between matches.

- Solving the optimization problem: the resulting binary linear programming problem is solved by dividing it into sub-problems if possible and solving each sub-problem using the branch-and-bound technique presented in [20].

- Disparity to depth conversion: using reconstruction equations the stereo matching results are used to generate a sparse depth map.

Experimental results and conclusions

The algorithm is able to compute depth values only along the contours that were matched. The problem associated to surface reconstruction from this map is not treated in [23]. The most straightforward solution to this problem is a linear interpolation between the depth values actually determined by the algorithm. The principal shortcomings of the algorithm is that it generates only a sparse depth map, which even though provides useful information, is not enough to obtain a complete description of the 3-D surface being inspected.

3.3 Inspection of Multi-Layered Integrated Circuits

A system for the inspection of multilayered wafer patterns is given in [13] and it constitutes an example of knowledge-directed processing. Rules and CAD data are used for defect detection and design verification.

Images to be processed by the system are fed to the processing unit in a raster-scan mode at video rate. Also a design pattern totally registered with the input image is fed to the control unit. The design pattern used is computer-generated lithography data. The positional information of the pixel of the design pattern indicates the layer membership of its counterpart pixel in the scanned image. Given the spatially synchronized images, the control unit generates suitable parameters (usually thresholds) specific to layer position. These thresholds are used in the segmentation of the patherns.

Normally there are 3 steps involved in the inspection of a product, namely candidate defects, "true" defect (false alarms suppressed), and fatal defect extraction. The last step involves some evaluation of true defects to see their degree of importance. The input and the output of all the stages are images (except the output of the last stage which can take other forms) so pipelined image processing seems most appropriate.

Defect candidate extraction

The knowledge for this stage has the following form:

Knowledge I:

If the pixel scanned belongs to layer i, and its gray level is not within allowable thresholds then the pixel must be flagged as being part of a defect candidate.

Note that in this case the feature value of the pixel is simply its graylevel. This scheme has great advantage in that it introduces a natural way of adaptive thresholding, since thresholds are layer dependent, even within each layer position dependent thresholds are feasible.

True defect extraction

The resulting binary defect candidate image obtained contains several false alarms mainly due to the following 2 reasons: harmless texture grains and slight alignment discrepancies between layers due to registration imperfections during fabrication. To eliminate the effect of grains the following knowledge is used:

Knowledge II-1:

If a pixel is within a defect candidate on layer i, and if the size of the candidate is less than τ_i, then the candidate is a grain image and the pixel should be deleted from the defect image.

A typical case arises when circular-shaped small defects (supposed to be grain image) must be deleted and elongated candidates must be left unchanged. Note again that the allowable grain size is adaptive in the sense that depending on each layer a suitable one can be generated by the control unit.

To eliminate the false alarms caused by imperfect registration between layers, it should be noted that discrepancies are likely to occur in the boundaries of the layers, so the following knowledge is employed:

Knowledge II-2:

If a pixel is within a defect candidate on the border of the layer i, and if the width of the candidate is less than p_i pixels, then the candidate is due to the pattern registration tolerance and the pixel should be removed from the candidate blob.

It should be noted that both processes for false alarm suppression can be executed in parallel, and the outputs combined into a single true defect image.

Fatal defect classification

The input to this stage is the true defect image and the output can take any desirable form like a list of judgment results. Many types of knowledge can be defined, one is given as a matter of example:

Knowledge III-1:

If a pixel is on the CAD pattern having a width W of less than a certain value, and if the actual pattern width w is less than q% of that of the corresponding design pattern W, then the pixel is a portion of a (semi of fully) open-circuited region.

To obtain statistical information on the wafer, a defect classification circuit can be used. This circuit partitions the patterned area of the wafer and measures such features as area, horizontal and vertical projected lengths of each defect involved in each partitioned area. These feature values classified for each layer are stored in the result memory and after the completion of the scan are input to a computer. This method, unlike conventional methods which just detect and display defects, can indicate what kind of defects are most likely on what layers of the Integrated Circuit.

Implementation and experimental results

Using the above ideas an inspection machine for logic IC wafers has been built. The resolution is 0.5 μm/pixel. The minimum detectable size of the defects is about 1.0 μm ; detection rate is about 95%, false alarm rate is 1 for every 100 chips. No speed performance was given. This limits the comparison of the system with other wafer inspection systems. A minimum detectable defect size of 1.0 μm may not be enough, since some common defects (tiny particles) can have sizes of the order of 0.5 μm or less and still be harmful. The two recent multilayered inspection systems by Yoda et al. and P300 can achieve a minimum detectable defect size of about 0.6μm, 0.5μm respectively.

3.4 P300: Memory Wafer Inspection

An automated inspection system for memory IC chips on multilayered wafers is presented. The heart of the algorithm consists of a self-reference algorithm which compares each pixel and its surrounding relevant pixels with its periodic counterparts. The idea is obviously applicable only to the inspection of repetitive scenes like memory cells. The method is based on a principle similar to the one used in [45] but there are some important differences.

The system

The image signal is digitized at a rate of 10 megapixels per second and produces a 480 × 512 pixel digital image. The system has 3 operational modes: set up, inspect, and review. In the set up mode the operator adjusts the system parameters. The inspection operation is then carried out automatically and the suspected defects stored. In the review stage, the defects found are displayed to the operator. The final decision regarding the fatality of the defects is left to the operator.

The inspection algorithm

The algorithm assumes horizontal periodicity of the object under inspection, with period R, and compares each pixel with two pixels at distance R away in either horizonal direction. The algorithm consists of two parts: "low-level" and "high-level" algorithms. The low-level algorithm outputs the defects with false alarms, the high-level algorithm does some noise compensation and false alarm suppression to get "true defects"

The low-level algorithm

In order to compensate for the presence of noise and allow small registration errors a pixel and its four adjacent neighbors are compared with the left and right periodic pixels and their neighbors. The current pixel is C_0 and its left and right counterpart are called L_0 and R_0 respectively. The low-level algorithm consists of the application of a simple binary nonlinear operator (GNLO).

The high-level algorithm

Basically the high-level algorithm consists of repeated application of low-level algorithm followed by a kind of enhancement applied to the accumulated result. This has the effect of false alarm suppression and increases the detection reliability. The steps comprising the high-level algorithm are the following:

- The same scene is imaged N times.
- The low-level algorithm is applied to each of these N images separately.
- An accumulator image is constructed in which each cell has a value between 0 to N, indicating the number of times the corresponding pixel in the scene was detected by the low-level algorithm to be a defect (a value of 1) or not (a value of 0). It is assumed that the random noises do not appear in successive imaging and application of the low-level algorithm, so the cells with higher counts can be thought as corresponding to true defect image. The simplest form of obtaining the true defect image is to apply a thresholding operation on the accumulator image.

Implementation and Experimental results

The high-level algorithm is performed in software but computationally intensive low-level algorithm is supported by dedicated hardware. The performance results given below are obtained using normal configuration of the P300 system which is thought to "optimize" the trade off between the 3 fundamental factors in an inspection system, namely detection probability, false alarm rate, and throughput. Results were obtained from testing a set of wafers having 2400 known defects. Setting the smallest detectable defect size to 0.5 μm, the detection rate is about 96% and reaches 100% as the defect size rises to about 1.5 μm. False alarm rate is 0.05% per frame (based on 226,488 frames). The system throughput is 45 seconds/mm^2. The system of Yoda et al. is about an order of magnitude faster, with comparable detection and false alarm rates (but it obviously has higher cost due to specialized hardware to perform morphological operations, CAD data manipulations, etc.)

3.5 LSI Wafer Inspection

A sub-micron defect detection algorithm for LSI wafer patterns has been developed in [26]. The essence of the method is a variant of a comparison technique in which two grayscale images are aligned by their detected edge patterns and compared by a new algorithm called Local Perturbation Pattern Matching (LPPM). The algorithm performs the matching by shifting one image in 8 plane-directions and in grayscale and finds the best match in a local window between the shifted image and its counterpart. The resulting unmatched regions are considered as defects. The method can detect defects of down to 0.3 $/mum$ in photoresist patterns.

Some other methods for inspection of single layer patterns operate on binarized images obtained after some preprocessing and subsequent thresholding. The authors in [26] argue that in order to be able to detect defects as small as resolution limits reliably it is necessary to use the grayscale image itself. Actually, gray-level based inspection of IC memory wafers was successfully used by two previous systems [12, 45].

Sensing

For precise automatic focusing a stripe pattern projected automatic focussing method [15] was applied. To prevent interference fringes caused by film thickness changes, a xenon lamp was used as the illumination device.

Algorithms

The concept of the algorithm is basically identical to a standard pattern matching with geometrical distortion, rotation of pattern, and size variation allowances. It is characterized, however, by the following two points:

- It utilizes the sign changes of the subtracted images for tolerating the error in normal patterns caused by the tiny shape differences or differences in the sampling position.
- It utilizes real-time processing with pipelined architecture, as all processing are executed with local operators in one path.

Local Perturbation Pattern Matching (LPPM)

When there is a tiny shape difference or an alignment error less than 1 pixel between two patterns, they are not matched perfectly by shifting the stored image by ± 1 pixel. But it is noticed that the sign of the subtracted image of the normal edge portion changes from positive to negative and vice versa by the ± 1 pixel shifting of the stored image, while the sign of the subtracted image of the defect does not change. Therefore one can eliminate the unmatched regions of less than one pixel by outputting a zero for the part where the sign of the subtracted image changes by a ± 1 pixel shift.

When there are allowable differences in grayscale between two patterns, the differences can not be eliminated by shifting the stored image in x-y plane. Then shifting of $\pm\alpha$ level in grayscale and the subtraction are also done in addition to the x-y shifting and subtraction operations.

From the 10 subtracted images thus obtained, a value of zero is output if the values of the corresponding pixels of the 10 subtracted images include both positive and negative values, or else, the absolute minimum value of the corresponding pixels is output as a defect signal. Finally the defect is detected by thresholding this output gray-scale image.

In IC inspection, it is normally recommended to sample the images at 1/2 to 1/3 the size of the minimum detectable defects. However, optical resolution is limited and sampling images with very small pixel size does not improve the resolution and increases the inspection time. To shorten the inspection time, the authors in [26] applied LPPM to images which are created by resampling the detected images with half-size pixels. Using this resampling technique defects as small as initial sampling pixel is detected and inspection time is improved by a factor of 4. Unfortunately, no precise information is given on how this general idea is actually used for the problem at hand.

Experimental results and conclusions

At 0.24 μ initial pixel size and 0.12 μ resampling resolution 0.3 μ defects on photoresist were detected. The number of tests carried out are not specifiedA apparently, the conclusions of the validity of the method are driven considering a few example tests. Sensitivity of the method to noise and other "bad" environmental conditions are not reported. No false alarm or efficiency rate is given.

Although not much detailed information is given in [26] on the actual digital defect detection algorithm, there is a strong resemblance between the proposed method and those used in other memory IC inspection systems [12, 45].

4 Packaging Inspection

4.1 Solder Ball Inspection in Integrated Circuits

In [6], the problem of analyzing images of solder balls in chip packaging is addressed. The images are grabbed using the shadows of solder balls obtained from an oblique illumination technique. As these shadows are cast on very complex and irregular appearance circuitry, the segmentation and defect detection tasks are difficult. Several possible methods are explored with special emphasis on feature-based automatic classification methods. An algorithm based on decision theoretic classification and feature extraction has performed well on available data. An architecture for fast implementation of the algorithm is also shown in the paper.

One of the popular chip packaging technologies is the so called controlled collapsed chip connection (C4). In this technique chips are placed directly on the ceramic substrate. To achieve this attachment, solder balls are placed on the chip surface and penetrate into the chip up to the last metalization level.

The quality of solder joints are given by good connectivity of the chip with the substrate which implies that enough solder must be present. This in turn imposes a predetermined volume, diameter and height constraint on the solder balls.

Several demands are imposed on a machine vision C4 inspector. The parameters to be measured involve the extraction of 3-D information. Also, high precision estimation of the parameters is necessary. Finally, computational efficiency is critical due to the large number of solder balls per chip (about 120) and large volume manufacturing.

As far as sensing is concerned, conventional methods are discarded, mainly because specular nature of solder balls saturates the captured images. In particular, bright-field imagery is useless since no 3-D information can be gathered successfully. Dark-field imagery has the same problem of not yielding 3-D information. In addition, the scattering response of the ball surfaces and some interference hinder the measurement of other parameters such as the diameter. On the other hand mechanical probing is not used because of its severe speed drawback. In order to image the solder ball array properly, an oblique viewing microscopy has been designed as a sensing mechanism (see [6] for details). Shadows are cast on the surface of the ball and the chip. Each shadow is formed by two overlapping ellipses. Considering the geometry of the optical setup with respect to the solder balls, it can be shown that from the separation of the centers and the minor diameter of either ellipse, an expression for the volume of the solder ball can be obtained. Hence, the detection and parameter estimation of the double elliptical shape is necessary to be able to derive an expression of the corresponding volume of the solder ball.

The proposed method

The first step consists of dividing the image into 121 overlapping subimages each containing a double-ellipse shape.

This simplifies the subsequent segmentation task and also allows possible parallelism. It also allows some "cleaning" of the subimages to be performed, since the largest object present in a subimage is the blob and the other spurious or noisy objects can be wiped out. Given the subimages, a segmentation technique based on extraction of multiple features for each pixel is used. The feature vector is then fed to a polynomial classifier to decide whether the pixel is a double-ellipse point or a background point.

To extract the subimages, the fact that blobs are aligned in the vertical and horizontal direction is exploited by taking horizontal and vertical projections of the original image. Separating lines between blobs are located by taking the local minima of the two projections.

Next, the segmentation of the double-ellipse is carried out, by working on each subimage separately. In this segmentation approach it is assumed that first of all pixels and their relationship with their neighbors (i.e. texture) can be described completely by a set of features. Secondly, different objects in a given scene are considered to be different in either gray level or texture or both. Hence, a set of features is computed per pixel and this vector is fed into a polynomial classifier that decides whether the pixel belongs to the solder ball or not. Some features used in a window are as follows: pixel gray level, pixel energy, mean gray level, energy in the window, local minimum, median value, local maximum, gray level variance, absolute value of the gradient, difference of the mean of the right and the left neighbors within the window, difference of the mean of the neighbors in 45 deg., difference of the mean of the neighbors in 90 deg., difference of the mean of the neighbors in 135 deg., value of the gray level histogram at the value of the center pixel, value of the histogram f_{15} at the value of the pixel, value of the histogram f_{16} at the value of the pixel, number of pixels in the window with gray level greater than the mean value, number of pixels in the window with gray level less than the mean value), etc. These features are calculated within a window of 5x5, and are implemented in hardware easily. See [6] for more details on the feature computation.

In the segmentation process, all the pixels that are not part of the solder ball shadow should be considered as a part of "the background". This can be accomplished by so called *supervised learning* which uses a polynomial classifier to decide which pixel belongs to the region of interest and which ones are in "the background". This approach consists of 2 steps:

- Training phase: The polynomial classifier is adopted by first interactively labeling the pixels of a training set and then adapting the parameters of the polynomial classifier based on this set of samples.
- Runing phase: The classifier is run over the real objects to be segmented and decision is made for each pixel if it belongs to the object of interest or to the background. For the case of two classes, a simple polynomial in the feature vector should be evaluated.

The authors used a subset of 18 subimages as a training set and labeling was performed by a tablet hooked up to an IBM 7350. A linear polynomial classifier was then adapted. The adapted classifier correctly classified 95subimages. Subsequently, all pixels of 18 C4 images (120 subimages each) were classified. The pixel labels were stored as images and as a final step the largest object in those labeled images were selected and slightly smoothed (by a 3x3 window operation).

Architectural issues

Aside from accuracy of the defect detection algorithm, efficiency is another key issue. In any practical system the "bottleneck algorithms" must be carried out by special hardware. To extract the subimages, an special pipeline architecture presented in [34] can be used.

In a first run over the image, the histogram processors calculate the corresponding histograms. In the next phase, the feature processors compute one feature each, and yield a feature vector register. This register is multiplied with one or more classification coefficients, and the results called "degree of object membership" are stored in memory. The final segmented image is obtained by taking pixel by pixel the maximum among the degrees of membership.

Conclusions

Although the method needs a large amount of features for each pixel, suitable hardware implementation permits the feature calculation in real-time. The proposed architecture is also claimed to perform pixel classification at video-rate speed. The proposed segmentation technique and architecture can be used in two different ways: to measure the volume of the solder balls, and for a accept/reject decisions. For the first application one needs a good criterion for determining the accuracy of the algorithm, one possibility is for example the use of some destructive SEM-based technique.

In more practical situations one may just be interested in a good/bad decisions. In this case, the training set containing good and bad solder balls is classified by a human operator, and later used to determine which features are appropriate for good/bad classification. These could be global shape characteristics obtained from the segmented images.

4.2 Inspection of Through-Holes in Printed Circuit Boards

In [3], an automatic though-hole inspection system for ultra-high density PWBs using leakage light detection has been developed. To cope with the increasing thickness of the PWBs, the sensitivity of the light detector is enhanced by a factor of 150 using a micro-channel plate tube. However, the tube caused two problems: stray light sensing and image distortion. The first problem is overcome by optically isolating the optics,and a distortion correction method is used to solve the second problem. With this system defects in through-holes as small as 100 μm is detected.

The authors previously developed an automatic optical through-hole inspection method called leakage light detection [2]. Masks block the entrance of light in the holes while the board is illuminated by bright light. The substrate transmits part of the light and if there is any defect on through-holes light leaks out and can be detected on the opposite side by a sensor. In order to implement these ideas a perfect masking system had to be invented and a discrimination algorithm devised to distinguish between leaked light and the light emitted by the substrate. Also with the advent of new ultra-high aspect-ratio PWBs much light is absorbed by the thick substrate (7mm or more) so a much more sensitive detection system must have been utilized.

Complete masking is accomplished by a special roller type mask which can handle board warpage of ± 1 mm. A CCD line sensor detects light from the unmasked hole position. The location signal is delayed by Dt which is the time required for displacement of the hole. A second sensor detects any light from the masked hole. The delayed location signal and leakage light signal are AND gated. Any defect on the through-hole will cause the output to raise indicating a defective hole. Sensitivity enhancement in the proposed method is accomplished using a special sensing arrangement. A micro-channel plate (MCP) tube is inserted in front of the CCD sensor. This arrangement intensifies the light over 150 times.

MCP tube causes two major problems: image distortion of up to 5% and stray light detection which lowers S/N ratio. The origin of the stray light was found to be the location light transmitted through unmasked holes. As both the location and leakage signals were imaged by the same lens, intense location light was becoming flare in the lens and was captured by ultra-high sensitivity sensor. To solve this problem the leakage light imaging and the position sensing optics are separated. Also a slit is inserted to block the light from substrate and adjacent holes from reaching the sensor. This arrangement eliminates the "noisy" light. On the other hand, the image distortion is corrected using an address correction method. The distortion of the leakage light signal is measured before with a resolution of 20 μm. The values are then stored and used later to adjust the image addresses. The resulting image is correct to within 0.5%.

4.3 Bond Pad Inspection

Electric testing of ICs using sharp probes on delicate bonding pads may scratch the surface of the pads and even the surrounding material. The authors in [1], developed a system called INSPAD. INSPAD checks the bonding pads for 3 types of common defects specified in the MIL-883C standards. Thresholding, morphological filtering and region extraction and analysis are the principal parts of the algorithm.

Pads are considered defect free if they fulfill the following criteria taken from MIL-883C standards:

- Probe marks must not extend beyond pad boundaries such that they damage glassivation. This kind of defect is refered to as a protrusion.
- Scratches on the bond pads must not exceed 50% of the pad width.
- The probe marks must not exceed 25% of the bond pad area.

INSPAD algorithm

It is assumed that the sensing system is already aligned with the bond pads. The algorithm uses simple image analysis techniques. To achieve defect candidate identification, a thresholding technique is used in which the pixels are sorted according to their gray level intensity and totals are calculated for the sum of 20% brightest and 20% darkest pixels, and the threshold is chosen as the average of these two sums.

Defect identification is performed by eliminating noise (false candidates) from the thresholded subimages (each subimage contains a bond pad). This is achieved by applying morphological operations using a 3 x 3 unit window as the structuring element. Once the "true" defects are identified, region extraction is performed and the properties of the defective regions are calculated. These include features such as area of the probe marks. To check for the protrusions, a window of a given tolerance is checked around the borders for any probe mark protrusions.

Experimental results and conclusions

The images are obtained from color polaroid photographs from an optical microscope. The computer used is an APOLLO DN-4000 workstation.

Just 37 sample bond pad images are used, with this small set of samples 100% detection rate is reported. Also the algorithm proved robust in the presence of noise, illumination variation between different images, illumination gradient within the same image, different bond pad sizes and shapes. The efficiency is reported as 2 to 3 seconds per bond pad. No false alarm-rate is reported.

5 Disk-Head Inspection

The inspection of finished air-bearing surface of disk heads is done by human operators using microscopes. The relatively large surface area of inspection (3mm $\times$ 4mm) and very small defect sizes (a few μms) makes the operator inspection a very tedious and time consuming task. Also the accuracy of the inspection is inconsistent and is inversely proportional to the degree of the fatigue or boredom of the human inspector. Furthermore, human operators are unable to properly verify dimensional characteristics of disk heads.

5.1 Automated Visual Inspection of Disk Heads

In [36], the problem of digital visual inspection of thin film disk heads is considered. orting As far as resolution is concerned the authors in [36] propose 2μm/pixel as sufficient for defect detection. This implies that if only one image has to be taken, a wide lens is necessary, and a 2K $\times$ 2K image have to be dealt with. As in the prototype stages conventional hardware is used, several images has to be taken in order to cover the whole scene. Furthermore, as defects have great differences in illumination responses, dark field and bright field microscopy is used which doubles the number of images to be taken, hence the efficiency of the inspection algorithm has paramount importance.

Because of the small number of defective samples available, the existence of position-based defects, and the need for some local shape analysis, conventional statistical classifiers are not suitable. Instead, a rule-based approach was used.

Algorithms

As pipeline architectures seemed promising, algorithms amenable to raster scan implementation were searched for. The main algorithms consisted of the following stages:

Shading correction

Due to the non-uniformities in the response of the camera, some shading effects were noticed. After linearizing the response of the system to the light source, we can assume a linear pixel by pixel model for shading correction of grabbed images. The following formula expressed the applied correction:

$$I_1(x,y) = (I(x,y) - B(x,y))C(x,y) \ \forall pixels(x,y)$$

where B and C are the "bias" and the "correlation factor" images respectively; I is the input and I_1 the corrected image. The bias image is obtained by closing the camera cap and grabbing a frame. Correction factors are obtained from digitized images of nominally uniformly reflecting or scattering surfaces for bright and dark field images respectively. The uniformity of these surfaces should be preserved after they are imaged by the sensing system. Obviously in the presence of shading effects that will not be the case. Details on the way correction factors can be computed in darkfield microscopy can be found in [36].

Boundary fitting and part location

Detection of boundary flaws involves some local measurement such as intrusion of a chip, and some global measurements such as width of the rail. Also, the extraction of boundary pixels is non trivial because of much noise and different textures of the surface. Rather, an ideal boundary is fitted to the noisy edge data based on *a priori* knowledge about the disk heads. A method such as Hough transform seems promising.

The authors in [36] reported a novel technique for computation of straight-line Hough transform in pipeline architectures. The outline of the algorithm is as follows: A coordinate reference image can be generated at any orientation:

$$R(i,j) = trunc(ai + bj) \ \forall pixels(i,j)$$

where a and b are constants. Further operations in the above equation can be approximated using look up tables, avoiding floating point computation and increasing efficiency. The image R represents a family of lines. This image and a binary gradient image δI are fed together to a histogrammer where:

$$\mathrm{Hist}(R(i,j)) = \mathrm{Hist}(R(i,j)) + \delta I(i,j)$$

is computed for each pixel. Both equations given above are computed in the raster scan mode and they can be pipelined since after R(i, j) is obtained, it can be used by the histogramming function. If these operations are realized in one stage of the pipeline the image δI can be forwarded to another identical stage where a different orientation is calculated.

Multicolor poligonal mask generation

A technique for generating multicolored polygonal masks in pipeline architecture is also presented in [36]. For more detailed description see [35].

Conventional graphics techniques to solve this problem in pipeline processors have strong drawbacks because these methods need random access of image planes, manipulation of pixel coordinates, and specialized raster-fill logic. The authors have implemented a novel method which overcomes the above limitations, and uses similar hardware resources as the Hough transform implementation.

The method to obtain coloring in pipeline architectures consist of two steps:

- Create a digital convex tessellation of the plane using the boundary lines found by the Hough transform, and those corresponding to inspection specified areas.

- Reconstruct polygons and segments of interest by assigning the same code to all pixels belonging to each of their fractions in the tessellated image.

Both steps can be accomplished in pipeline architectures by using the same coordinate reference generators, look up tables, and ALUs introduced for Hough computation. The final coloring of the polygon is obtained by a single look up table operation on the tessellated image. To accomplish this, one should know a priori the codes in the tessellated image which correspond to the polygon of interest. This information can be computed off line.

Segmentation of bright field and dark field images

For each zone, bright field thresholding is performed by a simple algorithm based on pixel histograms. On the other hand, images and defects in dark field imagery are typically textured. Therefore some preprocessing is done before taking the histogram. Starting from a shading corrected dark field image, repetitively apply the following operations: local minima of the previous image in a 5 × 5 window, and, local average of the previous image in a 5 × 5 window. The first operation tends to cluster and expand the low pixel values, while the second avoids noise propagation, and reduces variances of each class. After about 3 iterations, a histogramming operation follows and a heuristic threshold detection is used.

Feature extraction and defect classification

For each segmented object geometrical features are computed. These features are size features (e.g. area, perimeter), position features (e.g. centroid), shape features (e.g. area-touchsize), and gray level features (e.g. mean gray-level). The defect classifier uses object size, shape, position, and gray-level features to decide on defect type. A careful combination of known pattern recognition approaches is used. The classifier was implemented based on rules. The condition part of the rules is a combination of features that characterize a defect class. The action part selects a class for the object.

The system, experimental results, and conclusions

The system consist of an X-Y table, bright and dark field optics, a microscope with COHU 5000 TV camera, a DeAnza image processor, an special pipeline image processor, called RIPS, and a Motorola 68000-based general purpose computer.

Extensive experiments were performed and the following conclusions were drawn:

- False positive rate was extremely low. Generally no critical defect was missed. However, miss-classification rate for voids and dirt particles was about 25%, but the same problem occurs with human inspection.

- Execution time is in the order of 3.5 to 7 minutes per disk head. It is estimated that if a better optics is used together with long pipelines for part location and color mask computation, and a more powerful RIPS-type architecture, process time can be reduced to a few tens of seconds.

6 Inspection of Metal-Based Parts

6.1 Inspection of Aluminum Castings

In [8], several possible approaches to x-ray image analysis of cast aluminum wheels are described. Emphasis is given to the segmentation task through extraction of local features and the subsequent pixel classification problem. Some experimental results are also reported.

Typical requirements for x-ray inspection of industrial parts, for example aluminum wheel castings, are 100% detection rate, minimum defect size detectability of 1mm in critical regions. Low false alarm rate because of cost constraint. In such parts defects are generally due to cavities caused by gas bubbles or shrink holes, they occur isolated or in groups. They appear in x-ray images as small light blobs since in these regions the material is slightly thinner.

Presently, the inspection task is carried out by human experts, for short periods (best performance) they inspect a piece in 15-90 seconds.

The experimental system

Generally any automated x-ray inspection system consists of 4 parts: piece handling mechanism , x-ray tube, sensor, and the image processor. The handling mechanism should allow a good repeatability of the objects position, typically ±2 mm or less. The focal spot size of the x-ray tube must be chosen according to the resolution desired. For 1mm flaw detection, a resolution of $0.4mm \times 0.4mm$ was found adequate. The sharpness of the image and the magnification depends on the distance of the tube to the object and object to the sensor. Two different sensors have been studied, a 14" image intensifier and a linear array of scintillator-photodiode elements which consists of 512 identical detector elements with 1mm pitch. The dynamic range is about 1000. The data are digitized as 12-bit values , which are later compressed to 8-bits. Image intensifiers are used because they are well-matched to the x-ray spectrum used for scanning aluminum parts, they have good spatial resolution and are more available in the market. However linear array of scintillator-photodiode elements have better dynamic range and have more quantum efficiency for harder x-rays.

Algorithms

Image preprocessing

Before segmentation, some preprocessing of the image is necessary to compensate for fixed pattern noise and the exponential characteristics of the x-ray attenuation process.

- Nonlinear gray value transform: This transform implemented as a look-up table operation establishes an approximate linear relationship between the gray values of the resulting image and the thickness of the penetrated object. This look-up table is a function of the parameters of the imaging system and the object under inspection.
- Linear shading correction: The following method is used to perform the correction. Before inserting the object into the x-ray device, a low intensity image and a high intensity image of a homogeneous target are produced. From these two reference images offset and gain correction matrices a(i, j) and b(i, j) are computed, assuming a linear relationship between x-ray intensity and grayvalues. Matrices a(i, j) and b(i, j) are used for a linear transformation of the gray values $g_t(i,j)$, with individual coefficients for each pixel: $g_s(i,j) = a(i,j) + g_t(i,j) * b(i,j)$

Image segmentation

Segmentation is studied from the view point of local feature extraction and subsequent pixel classification. In the following, the problem of two-class pixel classification is considered and used for segmenting areas of interest in aluminum castings.

A) Features from image subtraction techniques

Features such as gray level difference between an "ideal" object and object under inspection can be useful for classification. Such techniques suffer seriously due to very precise registration requirements and due to the fact that some mechanical tolerances do exist which may cause the image comparison technique to produce large false alarms. To eliminate these inconveniences, Decker [11] applied a flexible matching technique, where the image under test is warped prior to subtracting it from the model image. In addition the two images are band-pass filtered prior to warping and subtraction. The method, however, is computationally costly and can still fail if mechanical tolerances are large enough.

B) Features from linear filtering operations

Enhancement of the flaws and suppression of regular features of the image can be achieved by using linear filters. Two examples are: unsharp masking and convolution with a difference-of-Gaussians (DOG) kernel. The cutoff or band-pass frequencies respectively, must be matched to the average size of the defect. For noise reduction purposes it is recommended to work with large kernels. This technique works well with images which do not have sharp edges. Despite of this drawback, linear filtering can be quite successful. The reason probably is due to the fact that DOG filters seems to be quite close to the optimal linear filter for the type of the flaws usually encountered.

C) Features from nonlinear filter operations

Only median filters has been studied. A median filtered image can be subtracted from the original one yielding good flaw signal; but the success depends on shape and orientation of the mask used. The result is good if for median computation a set of pixels is used which lie on the normal to the local edge direction. One approach [24] could be to design filters according to the regular features encountered in a given region of the image.

D) Features derived from a local flaw model

In [42], [43] an approach of a parameterized structurally isotropic flaw model for cavities has been presented, where the parameters of the model are used as features. The features are combined linearly based on heuristic assumptions. For simple structures such as edges this method works well, but for a more complicated ones like corners or ridges this method does not have enough discriminating power. Therefore *a priori* knowledge about the location of regular structures (e.g. rim of the wheel) were proposed along with further processing. This enhanced method worked well with low-noise images but had difficulties in noisy ones.

E) Combination of orthogonal, isotropic local features on the basis of training sets

All the approaches discussed above suffer from a limited scope of application. A more reliable technique would be the explicit local feature selection and classifier design. Polynomial classifiers [7] have been studied by the authors where basis local features are combined linearly or quadratically to form discriminant features for each class k. This approach has been used successfully in a number of applications in industrial inspection (see for example [33]).

When selecting features for combination, it is desirable to have basis features that are isotropic and orthogonal. Zernike polynomials are used for aluminum casting inspection. Subsets of the 4 lowest order Zernike polynomials are used to compute discriminant features with linear or quadratic combination of the basis features. The discriminant features are then thresholded to obtain segmentation results. Experience shows that the linear combination of second and third order Zernike polynomials when thresholded appropriately, gave the best segmentation results.

Pixel classification

Various methods for local feature extraction in aluminum casting x-ray images were discussed in the previous sections. The simplest method for classifying pixels into given classes is by thresholding the discriminant feature image. In the two-class classification case, this thresholding assigns a label to the flaw pixels and other to the background. Here the selection of an adequate threshold is the major concern.

Another alternative is classification by blob expansion. It consists of detection of local maxima in a feature image followed by blob expansion process. Two variant of this process have been studied. In one case blob expansion is done on the DOG filtered feature image and in the other directly on the discriminant feature image. Experiments show that the first method gives superior segmentation results. The fatal detect detection then proceeds by obtaining features pertaining to defect-blob candidates such as area and average contrast which are sufficient to discriminate between allowable and fatal defects.

The prototype

An automated x-ray inspection system has been built for a car manufacturer. In this system, a linear filter is used to generate the feature image, a blob expansion process uses as seeds pixels which are flaw candidates. The thresholds used are context dependent.

Several hundreds of tests were carried out on different wheels. The detection of the simple casting voids was 92% which gives near 100% detection rate for bad wheels since these generally contain voids in clusters. False alarm rate was reported to be around 4% per wheel and can be lowered to 1%. However, efficiency rate of human experts is better. There are some defects that cannot be detected by the aforementioned methods; such is the case for a defect in the form of a very large blob so that the whole spoke is missing. This and other issues like dimensional variation verifications call for a higher level analysis.

6.2 Inspection of Rolled Metal Surfaces

In [32], a prototype of an automated on-line metal strip inspection system is described. The system can detect and classify surface defects of copper alloy strip, though it can be extended easily to steel strip inspection. Extensive morphological preprocessing and statistical and structural defect recognition is used. The image analysis is carried out in commercial modules.

A metal strip is generally a 0.3-2.0 m wide and 0.1-5.0 mm thick shiny metallic bond. Surface quality is presently inspected by human experts which suffer from well-known inspection inconsistency and other problems.

As many as 20 different surface defects such as spills, scratches and roll marks can be distinguished. The criteria commonly used to decide if a surface has fatal defects basically depends on types of the defects , maximum number of defects per surface area, and the total number of defects on the inspected strip. Also such factors as the customer and the use of the strip can play an essential rule in that decision.

Some digital visual inspection systems have been designed for this purpose, but they are generally able to detect candidate defects, "true" defect recognition and evaluation is done by human operators.

The computational requirements for strip inspection are severe. Typically, the strip is 1m wide and moves at the rate of 1.5 m/s, minimum defect size is in the order of 1mm; also both sides of the strip must be inspected, this amounts to 3 mega-pixels/second.

The system

Sensing

Halogen light sources and 1024 pixel line scan camera were used. In this configuration the viewing angle can be tuned to optimize the detection of the most critical defect types. For copper strips a viewing angle of 2.5 deg. was found optimum for the detection of longitudinal spills, while in the cold rolled steel inspection viewing angle of 5 deg. enhanced the detection of several defect types.

The illumination arrangements, aside from improving sensitivity to certain defects, discriminate between 2-D and 3-D defects. 2-D defects are generally seen darker than the background since the dimmer prevents direct specular reflection to the camera. On the other hand 3-D defects are generally characterized as having sloped parts, so they are seen much brighter.

Inspection algorithms

In order to find the low contrast regions, the segmentation algorithm uses knowledge about their expected dimensions, shape and orientation.

Segmentation

A kind of morphology-based dynamic thresholding is used to obtain separate background images for light and dark blobs by applying opening and closing operations to the image [39]. The images so obtained are used to determine the light and dark threshold levels for the original image. The gray levels between the thresholds are classified as background. In practice, material dependent thresholds are chosen as a fixed percentage above or below the light or dark background respectively. So the thresholds depend on the size , shape and graylevel of the associated blob.

Blob classification

A connected component analysis is performed on the segmented image that calculates size, shape and orientation feature for each blob. Structural information of defects is important to classify a candidate blob as a true defect. For example, a spill consists of several blobs situated close together and distributed in certain orientation. One solution to the structural defect recognition problem is to describe defect candidates and their model as semantic networks [41]. An "edit distance" is defined between the two semantic networks which measure the effort needed to transform candidate network to model network. The defect is classified into the class for which the minimum distance is obtained [14].

The structural model described above is not fully satisfactory for all defect types. In addition another method is used which groups nearby blobs in a window and calculates features of the window and the included blobs. Some features are: width, height and area of the surrounding window, total area of the blobs, number of blobs in the window, and distance to the nearest neighbor group. This method enhanced the accuracy of the classification of certain defect types.

After classification stage the strip is accepted if the total number of defects per unit of surface is less than a threshold and no fatal defects are detected. Next, the quality class of the strip is determined considering types and density of the defects.

Implementation and experimental results

The image consists of 512 × 512 8-bit pixels. Resolution is about 1mm/pixel. Most image processing is done by hardware support. The system is realized by Max Video Data Cube Inc., Peabody, MA) and APA-512 (Vision Systems ltd., Adelaid, Australia) machine vision modules (Data Cube 87, Burford 87). The CPU is based on the Motorola M68020 microprocessor. The system is pipelined, a control unit synchronizes the operation of the units.

As far as speed is concerned, the time consumed by the prototype system to inspect a 512 × 512 image is 450ms, even though it is estimated that the system can be up to 8 time faster if more morphology-oriented hardware is added.

The system was tested in the laboratory and in a cutting line of cold rolling copper mill. Several hundreds of copper and steel strips was tested which contain different types of defects.

The system showed low false alarm rate. The illumination scheme proved to cut down false classifications by half compared to a conventional bright-field imagery.

The classification of large defects such as scratches worked well with "statistical defect classification" technique, but small area defects such as spills were not classified correctly. Structural recognition technique greatly enhances the performance in these cases.

The detection rate for the case of the spills is about 90%. Some further conclusions show that:

- No false alarm rate is given.
- No clear detection rate for all classes (or the average) is given.
- The later versions should improve the detection rate, 90% seems a low rating for industrial inspection.
- It is not clear what is the average detection rate of the human experts in the case of metal strip inspection.
- 0.5 mega pixels per second is not good enough speed for real time inspection of the metal strips, according to the basic inspection requirements this rate must be increased to about 3 mega pixels per second.

7 Other Inspection Areas

7.1 Carpet wear assessment

Appearance retention is very important as far as carpet quality is concerned. So far this is judged subjectively using assessment of panels of judges. Subjective methods have serious consistency problems.

Wear in the carpet is principally manifested as surface roughness, pile flattening, shedding of fibers, and, loss of color and pattern definition. A method for plain, 100% wool carpet wear assessment using image analysis techniques is developed in [37]. The effect of the patterns on the carpet is not investigated. Four sets of different carpets were submitted to experiments.

Features used

Numerical features on the second-order gray level statistics on the first-order gray level difference statistics, and other features are studied. The usefulness of these features in the discrimination of the degrees of wear are analyzed. Some features varied significantly when carpets are submitted to 2 hours (for standard 9 and Kilmarnock) and 4 hours (for Cellini and pastel weave) (see [37]).

Gray Level Difference Method (GLDM)

A set of features based on absolute differences between pairs of gray levels has been calculated. Let $d = (\Delta x, \Delta y)$ be a displacement, define $f^{'}(x, y) = | f(x, y) - f(x+\Delta x, y+\Delta y) |$. Let $P^{'}$ be the probability density function of $f^{'}$. With Ng gray levels this takes the form of Ng-dimensional vector whose ith component is the probability that $f^{'}(x, y)$ will have value i. One can compute $P^{'}$ from f by counting the number of times each value of the $f^{'}(x, y)$ occurs.

It can be seen that the coarseness of a texture can be measured from the distribution of values in $P^{'}$. If the texture is coarse and d is small compared to texture element size, $f^{'}(x, y)$ is usually small and the values in $P^{'}$ will be concentrated at $i = 0$. On the other hand for fine textures and d comparable to element size $P^{'}$ shows much wider spread. Several features extracted from $P^{'}$ are found useful, such as contrast, angular second moment, entropy, etc. [37].

Gray level Run Length Matrices (GLRLM)

Let $P(i, l)$ be the number of runs of length l in some given direction θ_0, of gray level i, consisting , in this case, of points whose gray levels lie in the range 0-20. A number of discriminant features are defined based on this function, considering that coarse textures tend to have many long runs while fine textures are characterized by having a higher proportion of short runs.

Neighboring Gray Level Dependence Matrix (NGLDM)

NGLDM is calculated considering the gray level relationship between each pixel and its neighbors at a distance d_0. This is a $Ng \times N_T$ matrix where N_T is the possible number of neighbors to a pixel. For a given d_0, and a threshold T_0, NGLDM can be evaluated by counting the number of times the absolute value of the difference between each element's gray level and its neighbors is less than or equal to T_0. Obviously features extracted from NGLDM are isotropic (for simplicity NGLDM ia denoted by Q).

The distribution of Q reveals some information on the coarseness of the texture. The election of adequate d_0, T_0 is problem dependent. Some useful features extracted from Q are suggested (see [37] for details).

Experimental results and conclusions

The sensing device used is a Hitachi color camera VK-C1500E, which has an iris and a solid state MOS sensor. 512 x 512 x 8-bit monochromatic images are generated using the matrox MIPS 512M image processor. The computer used is a Vax 11/750.

The most adequate illumination arrangement to grab images of good contrast, is to use four 100 Watts lamps placed at the corners of a 700mm x 450mm rectangle and at an angle of 28 degrees relative to the plane of the carpet.

The carpets were submitted to artificial wear using a Hexapod Tumber Tester. For the Kilmarnock and Standard 9 carpets, six images were captured from samples of carpet exposed to 0, 0.25, 0.5, 1.0, 1.5, 2.0, 2.5, and 3.0 hours of test.

For Cellini and Pastel Weave six images were captured from samples of carpets exposed to 0, 1, ...,6 hours of test. For each feature and carpet sample, absolute percentage change in feature values with respect to unworn control sample were plotted against duration of the test.

The analysis of the results revealed the following:

- Generally there are 3 stages in the wear process.
- Different features have different distinctive power on different types of the carpets.
- Some of the features belonging to the same class are found to be strongly correlated. Only a representative feature from a correlated set is used.
- Features extracted from NGLDM were found to have strong classification power.
- It is believed that this kind of image analysis techniques for carpet assessment is superior to human judges.
- This method can be applied basically in two areas: carpet grading and quality control in carpet manufacturing.

References

[1] M. Ahmed, C.E. Cole, R.C. Jain, and A.R. Rao. INSPAD: A system for automatic bond pad inspection. Technical report, AI Laboratory, The University of Michigan, June 1989.

[2] M. Ando, K. Mita, and T. Inagki. Automatic optical through-hole inspection method for printed wiring boards using leakage light detection. In *IEEE International Conference on Robotics and Automtion*, 1985.

[3] M. Ando and I. Takefumi. Automatic optical inspection of plated through-holes for ultralight density printed wiring boards. *Machine Vision and Applications, An International Journal*, 1(3), 1988.

[4] S.L. Bartlett, P.J. Besl, C.L. Cole, R. Jain, D. Mokherjee, and K.D. Skifstad. Automatic solder joint inspection. *IEEE Transactions on PAMI*, 10(1), January 1988.

[5] P.J. Besl, E.J. Delf, and R.C. Jain. Automatic visual solder joint inspection. *IEEE Journal of Robotics and Automation*, 1:42, 1985.

[6] W.E. Blanz, J.L.C. Sanz, and E.B. Hinkel. Image analysis methods for solder ball inspection in integrated circuit manufacturing. *IEEE Journal of Robotics and Automation*, 4(2), April 1988.

[7] W.E Blanz, J.L.C Sanz, and D. Petkovic. Control-free low-level image segmentation: Theory. In J.L.C Sanz, editor, *Advances in Machine Vision*. Springer Verlag, 1989.

[8] H Boerner and H. Strecker. Automated X-ray inspection of aluminium castings. *IEEE Transactions on PAMI*, 10(1), January 1988.

[9] R.T. Chin. Automated visual inspection: 1981 to 1987 survey. *Computer Graphics, Vision and Image Processing*, 1988.

[10] A.M. Darwish and A.K. Jain. A rule-based approach for visual pattern inspection. *IEEE Transactions on PAMI*, 10(1), January 1988.

[11] H. Decker. A Difference-Technque for automtic inspection of castings parts. *Pattern recognition Lett*, 2:125, 1983.

[12] B.E. Dom, V.H. Brecher, R. Bonner, J.S. Batchelder, and R.S. Jaffe. The P300: A system for automatic patterned wafer inspection. *Machine Vision and Applications, An International Journal*, 1(4), 1988.

[13] M. Ejiri, H. Yoda, H. Sakou, and Y. Sakamoto. Knowledge-directed inspection for complex multi-layered patterns. *Machine Vision and Applications, An International Journal*, 1990. To appear.

[14] M.A. Eshera and K.S. Fu. A graph distance measure for image analysis. *IEEE Transactions on systems, Man and Cybernetics, SMC*, 14(3):398, 1984.

[15] S. Fushimi and et. al. Automated visual inspection system for aluminium patterns on LSI wafers. In *Kodak Microelectronics Seminar Interface*, 1985.

[16] Y. Hara and et. al. Automatic visual inspection of LSI photomasks. In *5th. International Conference on Pattern Recognition*, page 273, 1980.

[17] Y. Hara and et. al. Automatic inspection system for printed circuit boards. *IEEE Transactions on PAMI*, 5(6), November 1983.

[18] Y. Hara and et. al. Automation of visual inspection of printed circuit board. *Electronics and Comunications in Japan*, 68(2):1, 1985.

[19] Y. Hara, H. Doi, K. Karasaki, and T. Tida. A system for PCB automated inspection using fluorescent light. *IEEE Transactions on PAMI*, 10(1), January 1988.

[20] F.S. Hillier and G.J. Lieberman. *Introduction to operations research*. Holden-Day Inc., 1989.

[21] M. Juha. Automated inspection of SMD solder connections. Technical report, IRT Corp. Publ., San Diego, CA, 1985.

[22] M. Juha. The economics of automated X-ray inspection for solder quality. Technical report, IRT Corp. Publ., San Diego, CA, February 1986.

[23] A.E. Kayaalp, A.R. Rao, and R. Jain. Scanning electron microscope based stereo analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern recognition*, June 1989.

[24] R. Klatte. Automatic detection of defects in castings by processing the context information of X-ray images. In *10th. IMEKO, Prague*, 1985.

[25] J.R. Mandeville. Novel method for analysis of printed circuit images. *IBM J. Res. Develop.*, 29:73, 1985.

[26] Y. Matsuyama, H. Iwata, H. Kubota, and Y. Nakagawa. Precise visual inspection algorithm for LSI wafer patterns using grayscale image comparison. In *IAPR Workshop on Computer Vision*, October 1988.

[27] W.E. Mc Intosh. Automating the inspection of printed circuit boards. *Robotics Today*, page 75, 1983.

[28] O. Mohtadi and J.L.C. Sanz. Technical Report: Recent advances in industrial machine mision. *To appear*, 1990.

[29] Y. Nakagawa. Automatic visual inspection of solder joints on printed circuit boards. In *SPIE*, volume 336, page 121.

[30] Y. Nakagawa, Y. Hara, and M. Hashimoto. Automatic visual inspection using digital image processing. *Hitachi Rev.*, 34(1):55, 1985.

[31] Y. Nakagawa and T. Ninomiya. Structured light method for inspection of solder joints and assembly robot vision system. In *1st International Symp. of Robotics Research*, Bretton Woods, NH, August 1983.

[32] T. Piironen, O. Silven, M. Pietikainen, T. Laitinen, E. Strommer, and M. Elsila. Automated visual inspection of rolled metal surfaces. *Machine Vision and Applications, An International Journal*, 1990. To appear.

[33] J.L.C Sanz, editor. *Advances in Machine Vision*. Springer Verlag, 1989.

[34] J.L.C Sanz and I. Dinstein. Proyection based geometrical feature extraction for computer vision : Algorithms in pipeline structures. *IEEE Transactions on PAMI*, 9:160, 1987.

[35] J.L.C Sanz, I. Dinstein, and D Petkovic. Computing multi-coluored poligonal masks in pipeline architectures and its aplicationsto automated visual inspection. *ACM*, 30(4), April 1987.

[36] J.L.C. Sanz and D. Petkovic. Machine vision algorithms for automated inspection of thin film disk heads. *IEEE Transactions on PAMI*, 10(6), November 1988.

[37] L.H. Siew, R.M. Hodgson, and E.J. Wood. Texture measures for carpet wear assessment. *IEEE Transactions on PAMI*, 10(1), January 1988.

[38] O. Silven and H. Hakalahti. A method for alligning printed circuit boards with design-data. In *4th Scndinavian conference on image analysis*, number 1, page 273.

[39] O. Silven and T. Laitinen. Methods for detecting blobs on non-textured surfaces. In M. Pietikainen and J. Roning, editors, *6th Scandinavian conference on image analysis*, page 1156, 1989.

[40] O. Silven, I. Virtanen, T. Westman, T. Pironen, and M. Pietikainen. A design data-based visual inspection system for printer wiring. In J.L.C. Sanz, editor, *Advances in Machine Vision*. Springer Verlag, 1990.

[41] O. Silven, T. Westman, T. Houtary, and H. Hakalahti. A detect analysis method for visual inspection. In *8th. International Conference on pattern recognition*, page 868, Paris, France, 1986.

[42] H. Strecker. Automatic X-ray testing of castings : An approach based on a local feature operator and flexible image matching. In *6th International Conference of Pattern Recognition*, Munich, October 1982.

[43] H. Strecker. A local feature method for the detection of flaws in automated X-ray inspection of castings. *Signal Processing*, 5:423, 1983.

[44] A.C. Vanzetti, Traub, and J.S. Ele. Hidden solder joint defects detected by laser infrared system. In *IPC 24th Annu. Meeting*, page 1, 1981.

[45] H. Yoda, Y. Ohuchi, Y. Taniguchi, and M. Ejiri. An automatic wafer inspection system using pipelined image processing techniques. *IEEE Transactions on PAMI*, 10(1), January 1988.