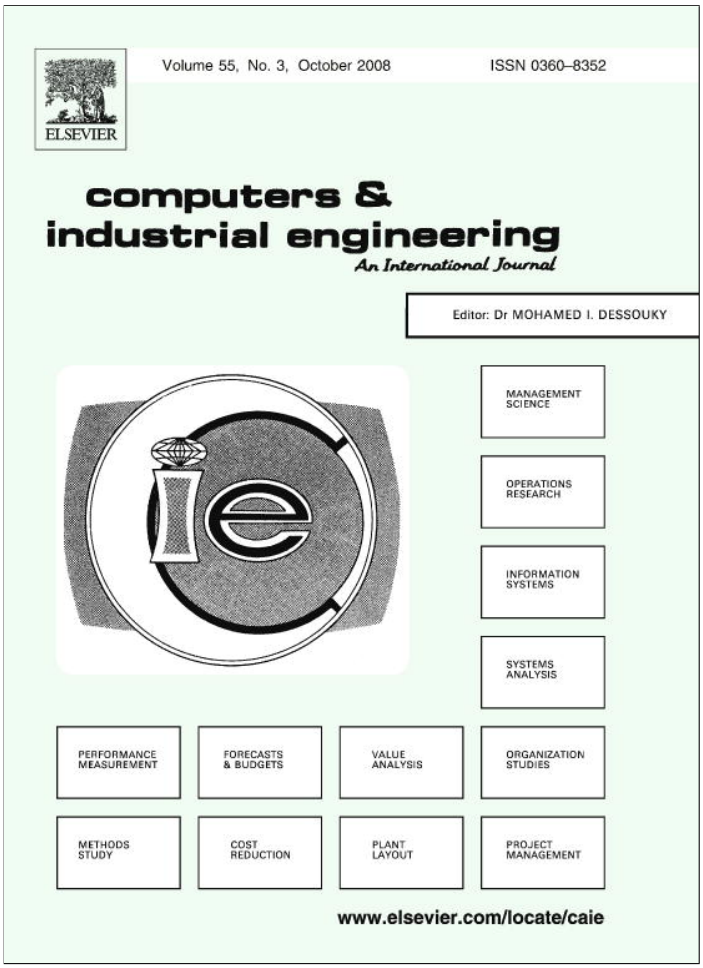

Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 55 (2008) 567–583
computers & industrial engineering
www.elsevier.com/locate/caie

An effective defect inspection system for polarized film images using image segmentation and template matching techniques ☆

Young-Geun Yoon [a], Seok-Lyong Lee [a,*], Chin-Wan Chung [b], Sang-Hee Kim [c]

[a] *School of Industrial and Information Systems Engineering, Hankuk University of Foreign Studies, 89, Wangsan-ri, Mohyeon-myon, Yongin-shi, Kyongki-do 449-791, Republic of Korea*
[b] *Division of Computer Science, KAIST, 373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea*
[c] *Key Technology Research Center, Agency for Defense Development, Jochiwongil 462, Yuseong-gu, Daejeon 305-152, Republic of Korea*

Received 8 March 2007; received in revised form 22 January 2008; accepted 22 January 2008
Available online 1 February 2008

Abstract

In this paper, we present an effective defect inspection system that identifies film defects and determines their types in order to produce polarized films for TFT-LCD (thin film transistor – liquid crystal display). The proposed system is designed and implemented to find defects from polarized film images using image segmentation techniques and to determine defect types through the image analysis of detected defects using template matching techniques. We extract features of the defects such as shape and texture, and compare them to the features of referential defect images stored in a template database. Experimental results using the proposed system show that it identifies defects of test images effectively (Recall = 1.00, Precision = 0.95) and efficiently (Average response time = 0.64 s), and also achieves a high correctness in determining the types (Recall = 0.95, Precision = 0.96) for five classes of defects. In addition the experiment shows that our system is fairly robust with respect to the rotational transformation, achieving the desirable property of the rotation invariance.

Keywords: Defect inspection system; Polarized film; Image segmentation; Template matching

1. Introduction

The rapid growth of information technology leads to the development of various display devices that play an important role for the visual interface between human beings and machines. Display devices such as

☆ This work was supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract UD030000AD.
* Corresponding author. Tel.: +82 31 330 4357; fax: +82 31 330 4093.
E-mail addresses: ygyoon@hufs.ac.kr (Y.-G. Yoon), sllee@hufs.ac.kr (S.-L. Lee), chungcw@cs.kaist.ac.kr (C.-W. Chung), falcon@add.re.kr (S.-H. Kim).

0360-8352/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2008.01.015

TFT-LCD, PDP (plasma display panel), and FED (field emission display), are being developed, replacing traditional CRT (cathode ray tube) devices. Among them, LCD leads related industry markets and has been used as an essential component in digital watches, mobile phones, laptop computers, and many other electronic devices. LCD is a thin and flat device that is composed of color or monochrome pixels arrayed in front of a light source. Each pixel consists of a column of liquid crystal molecules and a couple of polarized films. Their axes of polarity are perpendicular to each other. If the liquid crystals are not between them, the light passing through one would be blocked by the other. The liquid crystal twists the polarization of light entering one film to allow it to pass through the other. The molecules of the liquid crystal have electric charges on them. By applying small electric charges to transparent electrodes over each pixel or sub-pixel, the molecules are twisted by electrostatic forces. This changes the twist of the light passing through the molecules, and allows varying degrees of light to pass (or not to pass) through the polarized films.

The polarized film is one of essential components of LCD. In the process of attaching the film during manufacturing LCD, various defects such as bubbles, cunics, scratches, pieces of thread, dusts, and alien substances, might be produced. To detect those defects, it is necessary to use an automated vision system. In this paper we design and implement an effective defect inspection system for polarized films to enhance the productivity in manufacturing the LCD.

The defect inspection system is a device that extracts defect regions automatically and sends the recognition signal about them into the central controlling unit during the final step of film manufacturing process. The system plays an important role during the LCD manufacturing process since it prevents the possible malfunction by detecting defects timely and reduces time for identifying inferior products. The system consists of two components: the film image acquisition from polarized films and the image analysis for detecting defects. We focus on the latter, i.e. how to analyze film images to find defects correctly and efficiently.

The process to determine if a produced film has defects is composed of the following four steps: (1) A defect inspection system isolates defect regions on the film image using image segmentation techniques and issues an alarm signal by turning on a red-color lamp. (2) An engineer judges if they are removable (that is, in case they are pieces of thread or dusts). (3) If removable, one informs the other engineer to remove them. The engineer judges again after they have been removed. (4) If not removable, they are junked. It is needed that the above procedure is performed efficiently without stopping the production line. Most of existing methods detected defect regions successfully, but they need an engineer's intervention to judge if the defects are removable. If an automated system can determine defect types, the engineer's intervention would be eliminated or simplified, enhancing the efficiency of the whole process. In this paper we design and implement an effective defect inspection system to find defects as well as to determine defect types in polarized film images through image analyses. To determine defect types, we build a template database that stores the features of referential images such as shape and textures of defect regions. By comparing features of defects in a film image to the features of referential images stored in the database, the system is able to determine the types of defects effectively and efficiently.

The rest of the paper is organized as follows: Section 2 provides a survey of related work on defect inspection and image segmentation techniques. Section 3 presents the proposed defect inspection system with the image segmentation technique to detect film defects and to determine the types of the defects. In Section 4 we provide analyses and empirical results on the proposed system, and finally we give conclusions with future research directions in Section 5.

2. Related work

2.1. Film defect inspection system

An automated defect inspection system in polarized film images has been studied to find the defects in the TFT-LCD manufacturing industry since TFT-LCD became a primary product out of various display devices. However, there have been a few researches made even though it is an important and useful technology in the related field. Nakashima et al. (1994) classified defects into two classes: macro and micro. A macro defect is generated from the color inconsistency caused by the twist of panels. It is clear and big enough to be recognized by human eyes. Meanwhile a micro defect is so small that an engineer cannot detect it without the help

of a vision system. To detect micro defects the author proposed the subtraction method and the optical Fourier filtering. The former is used to detect black matrix holes or particles while the latter is used to find grain defects. Sokolov et al. (1992) developed a method that compares the histogram of intensity distribution of a defect image with that of referential images, to find defects during the final LCD inspection.

As a recent work, Lu and Tsai (2004) extracted defect regions using SVD (singular value decomposition). The TFT panel has orthogonal components in horizontal and vertical directions. The authors determined defect regions by eliminating the orthogonal components using SVD and re-building the image that contains the defect regions. Chang and Fu (1980) proposed a method that found defect regions using the transformed regression diagnostics and the threshold technique. The authors first isolated a region containing defects from a background using the transformed regression diagnostics, and then determined target defects from the binary image that had been generated from the threshold technique. The methods mentioned above were successful in finding film defects, but they have a problem in determining the type of defects, which is an important step in manufacturing TFT-LCD.

2.2. Image segmentation methods

The approach of image segmentation is categorized into four main classes: a threshold technique, a boundary-based method, a region-based method, and a hybrid technique that combine boundary and region criteria (Fan & Yau, 2001). A threshold technique is well performed when an image has two distinctive regions. But, if the image has ambiguous boundaries, the technique does not cope well with blurring at boundaries. Further, it neglects all of the spatial information of regions in the image. A boundary-based method depends on the assumption that pixel values change rapidly at the boundary between two regions. It detects edges using Sobel, Robert, or Canny operator, and utilizes those edges to identify boundary lines of regions. But those boundary lines identified by the method are only candidates of the boundary of objects, not exact boundaries, thus it must be combined with post-processing techniques, such as edge tracking, gap filling and smoothing. A region-based method is based on the assumption that neighboring pixels in a region have similar characteristics with respect to color, intensity, and texture. A well-known method is a split and merge technique (Chang & Li, 1994; Fan, Zeng, Body, & Hacid, 2005; Hijjatoleslami & Kittler, 1998; Shih & Cheng, 2005). In this method a seed pixel is selected, and starting from the pixel similar neighboring pixels are merged. This method shows satisfactory segmentation results, but the selection of an initial seed pixel influences the overall performance, and it is not simple to select the pixel. Pavlidis and Liow (1990) and Chu and Aggarwal (1993) proposed the hybrid techniques which integrate results of above approaches to provide a more accurate segmentation. Pavlidis and Liow (1990) performs the segmentation in such a way that it identifies regions using a split and merge algorithm, and then modifies or eliminates boundaries among regions, depending on the contrast with boundary smoothness and variation of the image gradient along the boundary. Chu and Aggarwal (1993) introduces a technique that integrates multiple region segmentation maps and edge maps from different channels. On the other hand, specific image segmentation techniques (Choi, 2003; Jain & Yu, 1988; Kim, Kim, & Choi, 2005) have been studied to extract the text contained in natural images using the intensity level. Such methods depend on the assumption that the text is distinctly divided from the background, which is a general characteristic of the text. But it may not be applicable for an image with a lot of noises and unclear objects. There are various region-based segmentation methods (Jianfeng, 2005; Kim, Chung, & Park, 2006; Passat et al., 2005) that handles unclear objects in the domain of medical imaging. Those methods however are mainly focus on utilizing the domain-specific features or knowledge, and thus may not be applicable for the film defect inspection.

The proposed segmentation method in this paper uses a hybrid approach which combines threshold and region-based techniques, and it is designed and implemented for monochrome images. While existing works select initial seed pixels and merge neighboring pixels of them, our method divides an image into blocks of a user-defined size, and determines the representative intensity of a block by averaging the intensity of pixels within the block. Then, an initial block is selected and neighboring blocks with similar characteristics are merged. The order of selecting blocks is from a center block to marginal blocks of an image. It is based on the experience that in most cases target objects are located in the center of an image. Whether neighboring blocks are merged or not is determined by a merging threshold. The proposed algorithm generates regions

using blocks instead of pixels and it eliminates pre-processing steps such as filtering and blurring that are adopted by most segmentation methods. It reduces noises since a block itself is represented by the average pixel intensity of the block. This simplification may speed up the process and reduce the possibility of the over-segmentation when a pixel is a merging unit.

3. The proposed defect inspection system

A vision system acquires film images which contain defect regions using various luminous sources and cameras, and analyzes them to control the film manufacturing process by finding defects. Fig. 1 shows the whole setup of our vision system for image acquisition, detection and classification of defects, display, recording, and reporting of defects. A line of eight cameras are used for image acquisition at different locations to detect different types of defects. We use CCD (charge coupled devices) line-scan cameras of 7K 100 MHz data rate and LED illumination sources. The acquired images are analyzed in the image analysis and inspection unit of the control rack to determine if film defects exist in the images. When a defect is detected, an appropriate signal is transmitted to the controller in the marking system to mark the place where it is found. Meanwhile, defect images are further analyzed for classification of defects according to the type of defects. An encoder is used to measure the speed of rotation of rollers and to transmit the film moving information to the data managing unit. It also provides the marking system with the location information of defects. The proposed defect inspection system is included in the the image analysis and inspection unit of the control rack.

As mentioned before, there are various types of film defects such as a piece of thread detached from workers' clothes, a cunic that is an extra-material between layers of polarized films, a pit in TAC (tri-acetyl-cellulose) or PVA (patterned vertical alignment), a bubble which is formed in the upper or lower layer during coating adhesives, and an alien substance that exists between films. In this paper our goal is to find all defects correctly and to determine the types of the defects found in polarized film images. Fig. 2 shows examples of defect images magnified 200 times by a microscope. In reality, the actual size of defects is within the wide range, from 50 to 1200 micra. It is sometimes difficult to recognize them by human eyes. Thus, we use high-resolution cameras to get clear and magnified defect images. The intensity of defect regions is different from that of non-defect regions since there is disperse and refraction of light around defect regions when the light penetrates into multi-layers of films (Jeong, Park, & Gang, 2003).

Image segmentation is a process which partitions an image into non-overlapping regions such that features in each region are similar while they are different in different regions. More formally, if F is a set of all image

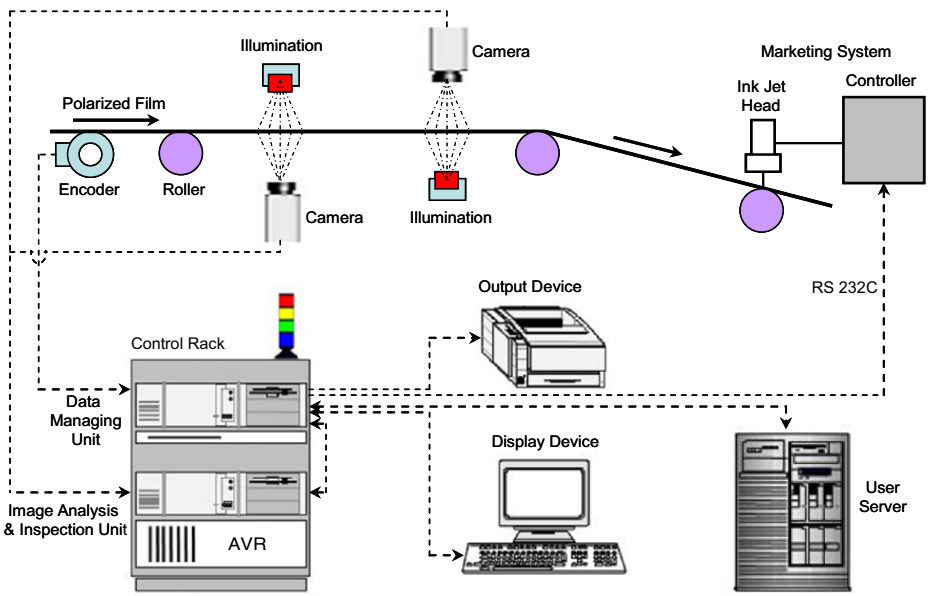

Fig. 1. The setup of the vision system.

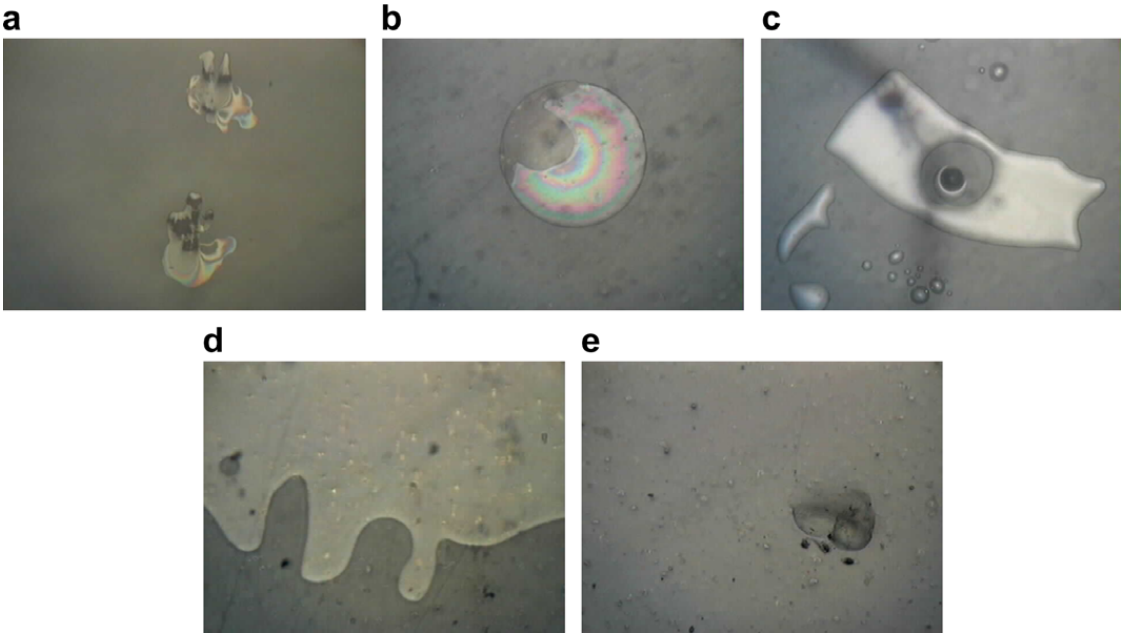

Fig. 2. Examples of defect images magnified 200 times by a microscope. (a) Cunic. (b) Alien substance. (c) Thread. (d) Bubble. (e) Pit.

pixels and $P(\cdot)$ is a homogeneity predicate defined over groups of connected pixels, then the image segmentation is a partitioning of the set F into a set of connected subsets or regions $\{S_1, S_2, \ldots, S_n\}$ such that

$$\bigcup_{i=1}^{n} S_i = F, \qquad S_i \cap S_j = \phi, \quad \text{for} \quad \forall i \neq j \tag{1}$$

The homogeneity predicate $P(\cdot)$ is such that $P(S_i) = true$ for all regions S_i and $P(S_i \cup S_j) = false$ for any two adjacent regions S_i and S_j(Cheng, 2001). The proposed method used in the defect inspection system is composed of the following four steps: (1) to generate a block matrix by tiling and 4-level quantization, (2) to generate defect regions by merging blocks using the segmentation technique, (3) to extract features from defect regions, (4) to determine the types of defects.

3.1. Generating a block matrix

An $M \times N$ image is divided into $m \times n$ blocks of size $p \times q$ ($1 \leqslant p \leqslant M$, $1 \leqslant q \leqslant N$) where $m = \lfloor M/p \rfloor$ and $n = \lfloor N/q \rfloor$. A right or bottom side of an image is not considered by algorithm if M/p or N/q is not an integer. The Eq. (2) computes the average intensity $I_{i,j}$ of block (i,j) ($0 \leqslant i \leqslant m-1$, $0 \leqslant j \leqslant n-1$).

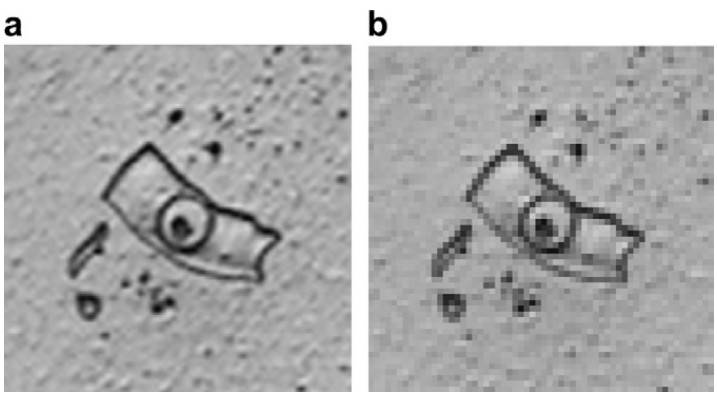

Fig. 3. An original image and its tiled image. (a) An original image. (b) A tiled image.

$$I_{i,j} = \frac{1}{p \times q} \cdot \sum_{h=pi}^{p(1+i)-1} \sum_{v=qj}^{q(1+j)-1} Hist(h, v) \tag{2}$$

In Eq. (2), $Hist(h,v)$ is a value of intensity histogram of location (h,v) of an image. Fig. 3 shows an original 180×180 image (a) and a tiled image with block-size 3×3. An average intensity value of each block is stored into a two dimensional array with size 60×60.

Most of natural images have a lot of noises, which requires a pre-processing step such as image filtering or blurring to remove them. But the tiling has advantage to eliminate overhead of this pre-processing step since the average intensity of pixels in block is used to represent the intensity of the block, which has an effect of removing noises. Each block of the tiled image can be represented by a quantized value according to its intensity. The number of these values are determined based on the characteristics of defect images. In our application, a block is represented by two bits such that a quantized value between level-0(00) and level-3(11) is assigned to it. We have performed extensive experiments using film images with various defect regions, and found that intensity values, 64, 128, and 196 are proper as splitting thresholds to distinguish *target* (defect) regions from *non-target* (background) regions. We name level-0 for a block with *avg_int* (an average intensity) $\geqslant 196$, level-1 for $128 \leqslant avg\_int < 196$, level-2 for $64 \leqslant avg\_int < 128$, and level-3 for $avg\_int < 64$. A two-dimensional array generated by tiling an image is called *block matrix* on which the post image analyses to find defect regions are based. A sample block matrix for the tiled image in Fig. 3(b) is presented in Fig. 4. When we define a block with value 0, 2 or 3 as a target block and a block with value 1 as a non-target block, our experimental observation on our test images has shown that most of defect regions are composed of target blocks, as depicted in Fig. 4.

3.2. Generating defect regions by merging blocks

The tiling step partitions an image into blocks of size $p \times q$ to process the image by $m \times n$ blocks instead of $M \times N$ pixels as merging unit. It is efficient since it reduces amount of information to be processed by $1/(p \times q)$ compared to the case of using pixels. The subsequent steps such as block merging and defect region generation consider target blocks only so that they could achieve the speed-up by eliminating non-target blocks from consideration.

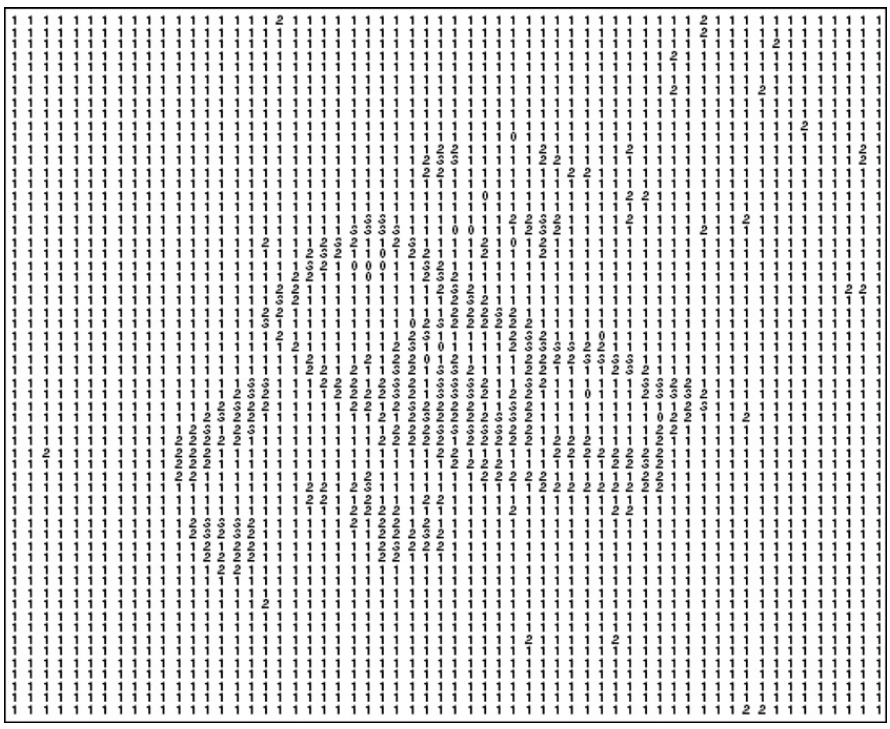

Fig. 4. A sample block matrix (60×60).

The order of evaluating blocks to merge blocks and generate regions is from central to marginal area of an image since target object regions are frequently located at the center of the image even though they are not always. When a target block has other target blocks at left, right, top, or bottom side of it, those blocks form adjacent target blocks. For example in Fig. 5, if an image has target blocks $\{1,3,4,6,8,12\}$ and non-target blocks $\{2,5,7,9,10,11\}$ then the defect region generation is performed as follows.

First, neighboring blocks are evaluated from target block 6 which is located at the center of image. The blocks which have been visited are excluded from consideration. In the example, region r_1 is generated with initial block 6, and adjacent blocks $\{5,7,2,10\}$ of block 6 are evaluated. But all adjacent blocks are non-target blocks, causing the merging process to be finished for region r_1. Next turn is block 7 to generate another region. Block 7 is however visited before, thus next block 11 is evaluated which is non-target block. Similarly, blocks 10, 9, 5 are evaluated which are all non-target blocks. Next block 1 is found to be a target block, generating new region r_2. There is no adjacent target block for block 1, thus the merging process for the region $r_2$is finished and moves to block 3. New region $r_3$is generated with block 3 and target blocks 4, 8, and 12 are added to r_3 since they are adjacent target blocks of block 3, 4, and 8, respectively. After all blocks are visited, the process of merging blocks and generating regions is over. An algorithm *Image_Segmentation* in Fig. 6 shows above procedure where a function deQueue(Q) picks a block from the queue Q and return it for the subsequent process. A block-merging procedure is described by a recursive algorithm *Merge_Block* in Fig. 7.

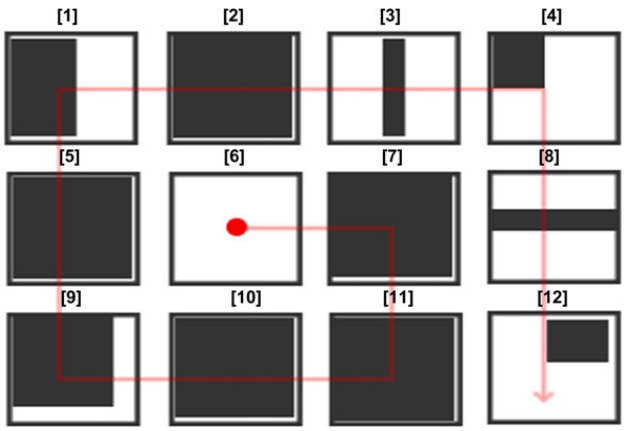

Fig. 5. An example of defect region generation.

Algorithm *Image_Segmentation*

```
Input:    an original image I, blocksize parameters p,q
Output:   a set of target regions R
Step1:    // tiling an image
          produce a blocked image BI by partitioning I into blocks of size pxq
Step2:    // produce defect regions by merging blocks
          R ← φ   // initialize a set of target regions
          queue Q ← a sequence of blocks to visit
          VISIT_FLAG of all blocks ← OFF
          blk ← deQueue(Q)
          while (blk is not NULL){
            if (VISIT_FLAG(blk) is OFF & blk is a target block) then
              initialize a region rgn
              rgn ← Merge_Block(blk, rgn)
              R ← R ∪ rgn
            end if
            blk ← deQueue(Q)
          end while
Step3:    return R
```

Fig. 6. An algorithm *Image_Segmentation*.

After generating multiple candidate regions, target defect regions are determined among the generated regions. There are various approaches to determine target defect regions depending on features of an image. We first merge regions if the distance between centers of two regions is within a predefined value, and then determine target defect regions among the merged regions when the sizes of them are higher than a predefined size. Fig. 8 shows multiple candidate regions from the 60 × 60 block matrix for the image shown in Fig. 4. Table 1 describes extracted candidate regions and their sizes.

```
Recursive Algorithm Merge_Block
Input:    BLOCK blk, REGION rgn
Output:   REGION rgn
Step1:    // check if blk has been visited and if it is a target block
          if (VISIT_FLAG(blk) is OFF) then
             VISIT_FLAG(blk) ← ON
             if (blk is a target block) then
                rgn ← rgn ∪ blk
                Merge_Block(RIGHT(blk), rgn)     // merge a right-side block of blk
                Merge_Block(LEFT(blk), rgn)      // merge a left-side block of blk
                Merge_Block(UPPER(blk), rgn)     // merge a upper block of blk
                Merge_Block(LOWER(blk), rgn)     // merge a lower block of blk
             end if
          end if
Step2:    return rgn
```

Fig. 7. A recursive algorithm *Merge_Block*.

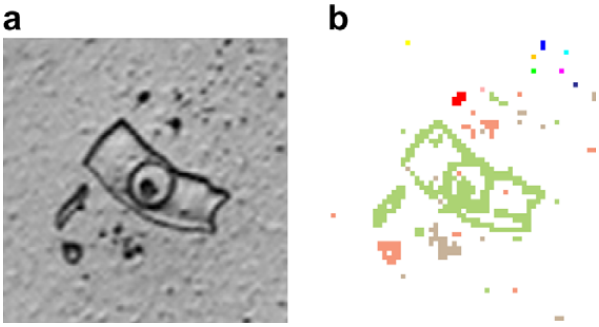

Fig. 8. Extracted candidate regions. (a) An original image. (b) Multiple candidate regions.

Table 1
Extracted candidate regions and their sizes

Region No.	Number of blocks	Region No.	Number of blocks	Region No.	Number of blocks
0	1	15	210	30	30
1	2	16	9	31	1
2	1	17	1	32	1
3	1	18	1	33	1
4	1	19	2	34	1
5	1	20	1	35	24
6	1	21	1	36	4
7	1	22	2	37	2
8	7	23	1	38	1
9	3	24	5	39	2
10	1	25	2	40	19
11	2	26	1	41	1
12	2	27	1	42	1
13	1	28	1	43	1
14	2	29	1	44	2

Final defect regions are determined by merging multiple candidate regions extracted. During this process a region with a few number of blocks (within a pre-defined threshold) is eliminated since it is regarded as a noise that is generated during the image acquisition. Merging regions is a suplementary process for the case that a single defect region is split into multiple candidate regions during merging blocks. Whether two regions are merged or not is determined by considering the placement of minimum bounding rectangles (MBR's) of those regions. When the MBR's of two regions are overlapped then those regions are merged. Fig. 9(b) shows that three separate regions are determined without region merging, while in Fig. 9(c) a single defect region is generated by merging regions. As we can observe in the example, merging regions leads to the better segmentation, causing meaningful defect regions to be identified.

3.3. Extracting features from defect regions

To represent the characteristics of a defect region we extract various features from it. Fig. 10(a) depicts a collection of MBRs for candidate regions shown in Table 1. Fig. 10(b) represents a final defect region with noises removed, where a black rectangle is an MBR of the defect region and a set of white blocks forms an actual defect region (ADR). We use four features to represent the characteristics of defect regions as follows: (1) the area ratio of ADR with respect to ADR's MBR (f_1), (2) the center of gravity of ADR with respect to ADR's MBR (f_2), (3) the contrast using texture features (f_3), (4) the ASM (angular second moment) using texture features (f_4). Among the features, f_1 andf_2 represent the approximate shape of a defect, while f_3 and f_4 represent the shape and texture characteristics of it. Those features are used to determine the type of the defect.

The area ratio of ADR with respect to ADR's MBR (f_1: *adr_ratio*) is defined as the number of blocks in ADR divided by the number of blocks in MBR, and is represented by Eq. (3):

$$f_1 : adr\_ratio = \frac{\text{of blocks in ADR}}{\text{of blocks in ADR's MBR}} \tag{3}$$

Fig. 9. Determining defect regions without/with region merging. (a) An original image. (b) Without region merging. (c) With region merging.

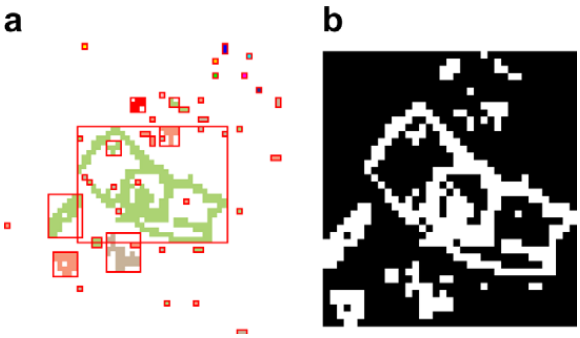

Fig. 10. An actual defect region (ADR) from multiple candidate regions. (a) Extracted candidate regions. (b) An actual defect region.

The center of gravity of ADR (f_2) is represented by a vector **C** in rectangular coordinates, which is (C_x, C_y), the relative coordinates of the ADR's center with respect to an origin of ADR's MBR, and expressed as Eq. (4):

$$f_2 : \mathbf{C} = (C_x, C_y) = \left(\frac{(\text{center of gravity of ADR})_x}{(\text{length of edge of ADR's MBR})_x}, \frac{(\text{center of gravity of ADR})_y}{(\text{length of edge of ADR's MBR})_y} \right), \tag{4}$$

Fig. 11 shows (C_x, C_y) for five defect types mentioned above by averaging all defects in our experimental sets. As the distance between defect types is getting longer in rectangular coordinates, they are clearly discriminated each other. As shown in the figure we can observe that most of defect types are clearly separated except for that of alien substances and bubbles. These two types are represented as 'similar' with respect to f_2 but the rest of the features compensate for that.

The texture features, f_3 and f_4, are determined by the contrast and ASM, respectively, using GLCM (gray level co-occurrence matrix) (Haralick, Shanmugam, & Dinstein, 1973), showing the variation and alignment of blocks in the MBR of a defect region. GLCM is a two-dimensional matrix which contains information on the frequency of gray-level variation between two adjacent blocks. We formed four matrices with respect to 0°, 45°, 90°, and 135° directions for those features to keep the rotation invariant characteristics. Consider a sample image with 4 gray levels $(0,1,2,3)$ and 4×4 blocks as shown in Fig. 12(a). A framework matrix in Fig. 12(b) is used as a reference matrix for representing gray-level variations. Then GLCM is formed as follows: To form GLCM for 0° direction, we consider the gray-level variation with respect to left-to-right and right-to-left directions in the framework matrix. The frequency of gray-level variation between adjacent blocks is 4 for $(0,0)$, 2 for $(0,1)$, 1 for $(0,2)$, 0 for $(0,3)$, 2 for $(1,0)$, 4 for $(1,1)$, 0 for $(1,2)$, 0 for $(1,3)$, 1 for $(2,0)$, 0 for $(2,1)$, 6 for $(2,2)$, 1 for $(2,3)$, 0 for $(3,0)$, 0 for $(3,1)$, 1 for $(3,2)$, and 2 for $(3,3)$, respectively, resulting in Fig. 12(c). For remaining 45°, 90°, and 135° directions, we form matrices similarly as shown in Fig. 12(d)–(f).

The GLCM is effective in representing the texture characteristics since it reflects the variation of gray-level values among blocks. For the rotation invariance we compute the respective contrast and ASM value from GLCM for 0°, 45°, 90°, and 135° directions and take an average of them. The contrast *cont* for feature f_3 and ASM *asm* for feature $f_4$are computed by Eqs. (5) and (6), respectively:

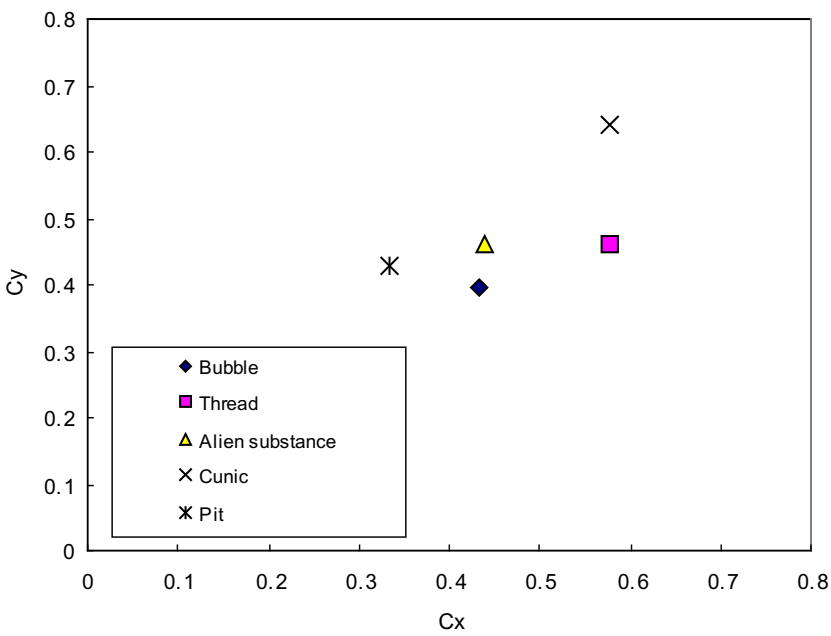

Fig. 11. The coordinates (C_x, C_y) for five defect types.

a

0	**0**	**1**	**1**
0	**0**	**1**	**1**
0	**2**	**2**	**2**
2	**2**	**3**	**3**

b

0,0	**0,1**	**0,2**	**0,3**
1,0	**1,1**	**1,2**	**1,3**
2,0	**2,1**	**2,2**	**2,3**
3,0	**3,1**	**3,2**	**3,3**

c

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

d

6	0	2	0
0	4	2	0
2	2	2	2
0	0	2	0

e

2	1	3	0
1	2	1	0
3	1	0	2
0	0	2	0

f

4	1	0	0
1	2	2	0
0	2	4	1
0	0	1	0

Fig. 12. The construction of GLCM matrices. (a) Image with 4 gray levels. (b) Framework matrix. (c) 0° direction. (d) 90° direction. (e) 135° direction. (f) 45° direction.

$$f_3 : cont = \underset{0^o,45^o,90^o,135^o}{avg}\left(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} P_{i,j}(i-j)^2\right) \tag{5}$$

$$f_4 : asm = \underset{0^o,45^o,90^o,135^o}{avg}\left(\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} P_{i,j}^2\right) \tag{6}$$

In Eqs. (5) and (6), $P_{i,j}$ is the frequency at cell (i,j) of GLCM, where i and j are indices for row and column of the matrix respectively. In computing f_3, a weight is to be considered, which is a square of difference between i and j, such that f_3 has a larger value when the variation of intensity between blocks increases. On the other hand, f_4 becomes larger when the distribution of frequency values within the matrix is changed rapidly. It is because ASM is computed by the square of frequency values. The uniform distribution of the values leads f_4 to be small. Thus f_4 serves as a measure for the directional property of an image. We examined contrast and ASM features to check if they have the rotation-invariant property, using Brodatz texture images in the USC-SIPI image database in University of Southern California (USC-SIPI Image Database), which has been widely used in related studies. We first made rotational transformations (0°, 30°, 60°, 90°, 120°, 150°, and 200°) on the images and then observed the change of contrast and ASM values. As shown in Fig. 13, those two values are almost consistent regardless of the transformation.

Fig. 14 plots the averages of contrast and ASM values for five types of defects mentioned previously. We observe in the figure that the difference among types is big enough to discriminate, thus we are able to use those features effectively to determine the types of defects.

3.4. Determining the types of defects

To determine the type of a defect we use the template matching method. A template database stores feature values for referential defect images, the types of which have been classified in advance. When a polarized film image is to be examined, the system first finds defect regions by generating a block matrix for the image and by isolating defect regions through block merging, and then extracts features from defect regions. To determine the types of defects it compares the extracted feature values to those stored in the database. When the system finds the match it determines the type of a defect. In this paper we use four features, f_1, f_2, f_3, and f_4, for determining the type. Our inspection algorithm that is embedded into the vision system in Fig. 1 is to classify five

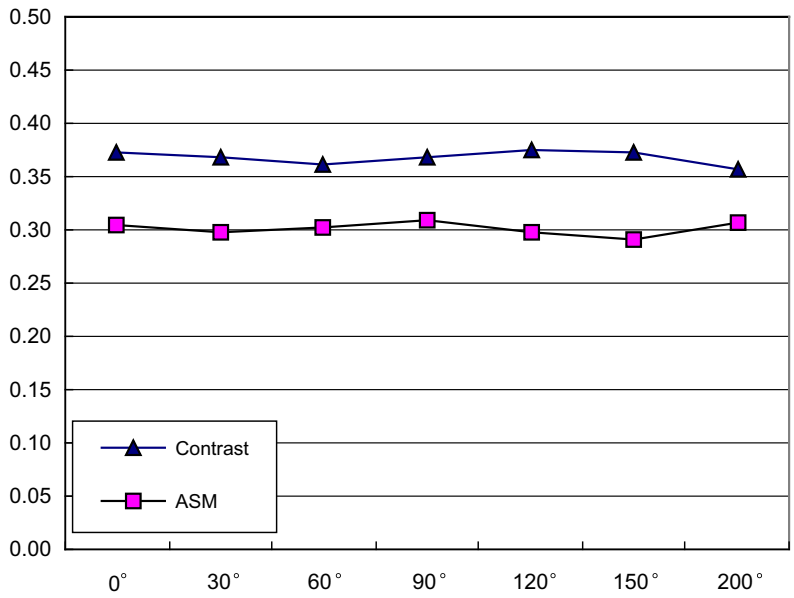

Fig. 13. The change of contrast and ASM values for rotational transformations.

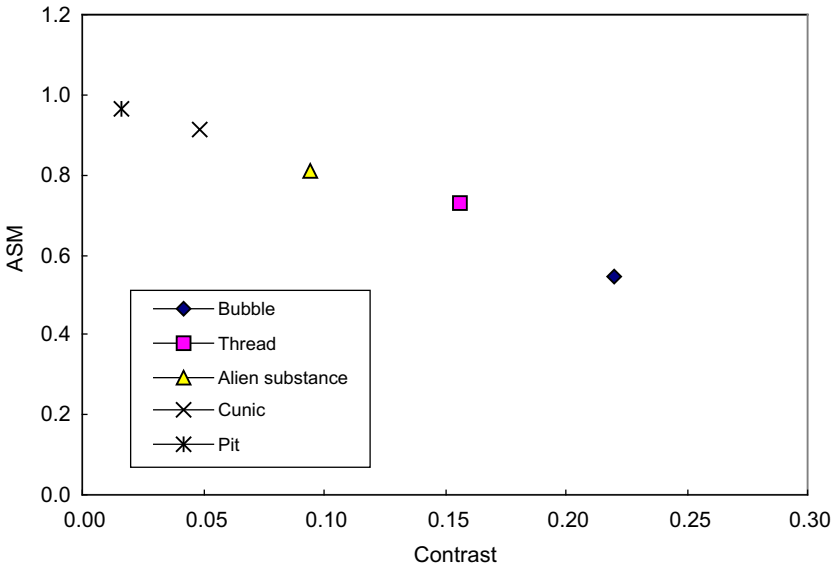

Fig. 14. Distribution of defect types with respect to *cont* and *asm*.

types of defects in Table 2 that occur very frequently in reality. To support other types of defects like MD scratches, gels, and other contaminations, different types of methods and devices are needed. Those works are planned as a future implementation. Table 2 shows the example of a template database.

Table 2
An example of a template database

TID	f_1	f_2		f_3	f_4	DESC.
T01	0.057	0.577	0.641	0.089	0.843	Cunic
T02	0.183	0.433	0.397	0.216	0.551	Bubble
T03	0.619	0.333	0.429	0.604	0.183	Pit
T04	0.235	0.579	0.459	0.366	0.451	Thread
T05	0.134	0.439	0.462	0.181	0.648	Alien substance

Determining defect types is done by computing the similarity between feature values extracted from defect images and those of referential images stored in the database. The similarity is represented by a distance function between two features of defect images and template images in the feature space. A similarity value between the two features becomes smaller as the distance between them increases. Defect types are determined by matching a defect image to a referential image whose distance value with respect to the defect image is the smallest among all template images in the database. The distance, $dist(R,T)$, between a defect region R and a template T in the database is computed by Eq. (7):

$$dist(R,T) = \sum_{i=1}^{n} w_i \cdot d_i \tag{7}$$

In Eq. (7), n, w_i and d_i are the number of features used, the weight ($\sum w_i = 1$), and the distance between R and T, respectively, with respect to a feature i. Users are allowed to change the weight w_i based on their application domain. The classification of defect types is quite subjective and thus different assignments of weight values may reflect different user requirements and preferences. The distance d_i with respect to each feature is computed by Eqs. (8)–(11):

$$d_1 = |adr\_ratio_R - adr\_ratio_T| \tag{8}$$

$$d_2 = (\mathbf{C_R} - \mathbf{C_T})^{\mathbf{T}}(\mathbf{C_R} - \mathbf{C_T}) \tag{9}$$

$$d_3 = \frac{|cont_R - cont_T|}{|cont_R + cont_T|} \tag{10}$$

$$d_4 = \frac{|asm_R - asm_T|}{|asm_R + asm_T|} \tag{11}$$

In above equations, d_1 represents the difference of ADR ratios for a defect region R and a template T in the database. $\mathbf{C}_R$ and $\mathbf{C}_T$ are vectors which represent relative coordinates of centers for R and T respectively, with respect to ADR's MBR, and thus d_2 is a Euclidean distance between those two vectors. d_3 and d_4 are the distances that normalize the differences of Contrast and ASM for R and T, respectively.

4. System implementation and empirical study

To evaluate the effectiveness of our proposed method, we implement a film defect inspection system using Java under the Window XP Server environment. The inspection system is included in the the image analysis and inspection unit of the control rack in Fig. 1, and is designed to find defects from polarized film images and determines the types of them. Fig. 15 shows a screen shot of the implemented system which displays a test query image with defects identified.

In above screen shot of the system, a left-upper window is an image viewer that browses a query image, and a right-upper window is a display area that shows the MBRs of defect regions extracted from the query image. Two windows are scrollable and shows parts of defect images. Thumbnails in a bottom section show a set of binary images for defect regions identified. The goals of our experiment are to show that: our system detects all defects from test images by region growing based image segmentation (Exp-1), and our system determines properly the types of detected defects (among cunic, bubble, pit, thread, and extraneous substance) by template matching (Exp-2), and our system is rotation-invariant for detecting defects and determining defect types (Exp-3). Table 3 summarizes the experimental parameters setting which is used for our empirical study.

Test images are acquired from the vision system and selected among images that include five types of defects. Each image includes one or more defects, and our test images have total 145 defects (38 bubbles, 29 threads, 32 alien substances, 20 cunics, 25 pits, and one unknown defect) as shown in Table 4. The experiment also has been done for artificially transformed images in which the defects are rotated in a 30-, 60-, and 90-degree arc, respectively, to show that our method is robust with respect to the rotational transformation (rotation-invariant). The block size is selected to be 3×3 pixels considering the size of usual noises and defect regions in reality. During determining defect regions, the regions that contain fewer blocks less

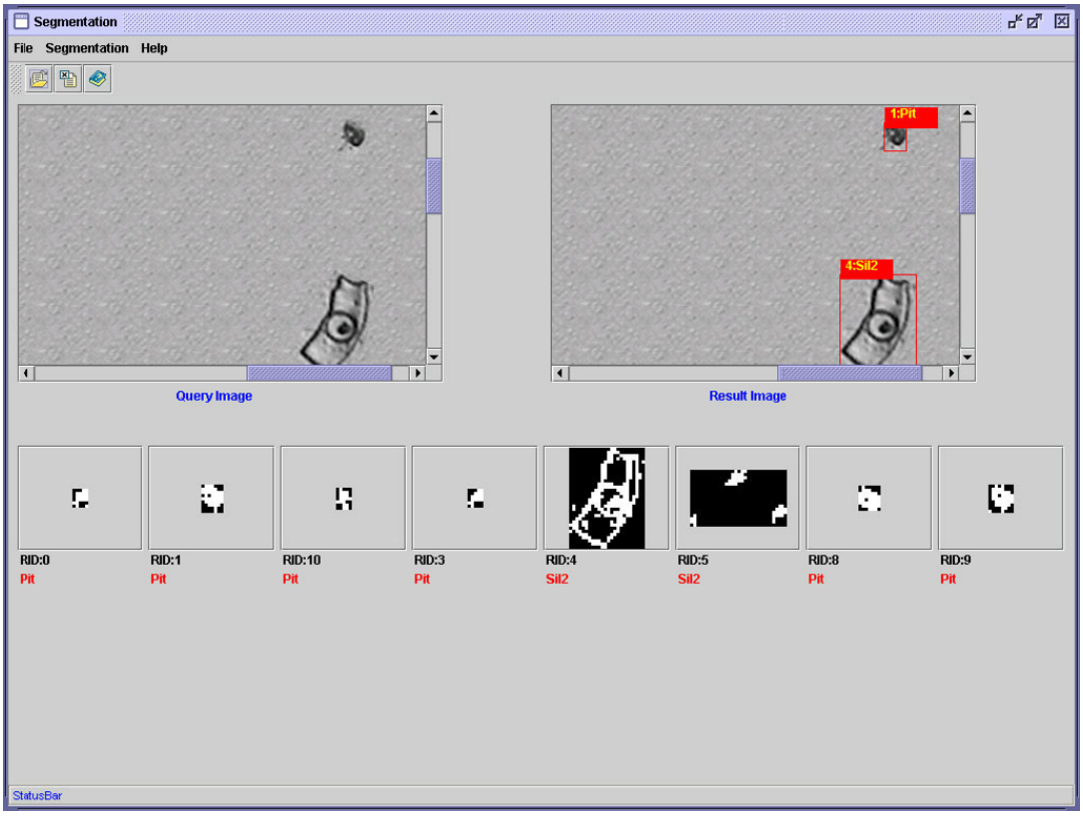

Fig. 15. A screen shot of the implemented system.

Table 3
Experimental parameters setting

Parameters	Setting values/descriptions	Remarks
Test images	45 large-size images (1000 × 1000 pixels)	Including 5 types of defects (Refer to Table 4)
Image rotation	30-, 60-, 90-degree	For ration-invariance test
Block size	3 × 3 pixels	
Minimum region size	5 blocks	Threshold for region size
Number of features	4 (f_1, f_2, f_3, f_4)	Refer to Section 3.2
Feature weights	0.35(f_1), 0.15(f_2), 0.25(f_3), 0.25(f_4)	Refer to Section 3.4

Table 4
A confusion matrix determining defect types

Defect type		Predicted type					Total
		Bubble	Thread	Alien substance	Cunic	Pit	
Actual type	Bubble	38	0	0	0	0	38
	Thread	0	29	0	0	0	29
	Alien substance	1	0	31	0	0	32
	Cunic	0	3	0	17	0	20
	Pit	1	0	0	0	24	25
	Unknown	0	1	0	0	0	1
Total		40	33	31	17	24	145

than five are eliminated from consideration since they are regarded as noises. We extract feature information ($f_1 \sim f_4$) from defect regions of referential images and establish a template database that stores the feature information as shown in Table 2. The database is used to determine types of defects within a query image

by comparing feature values of them to the values in the database. The weights to compute the similarity between defects are determined empirically so that they could maximize the difference among defect types. As a result the weights for f_1, f_2, f_3, and f_4 are determined to be 0.35, 0.15, 0.25, and 0.25, respectively.

To evaluate our method, we have used the precision and recall that are well known in the information retrieval and classification applications. Let *Set*(*pred*) be a set of predicted objects classified by our method, and *Set*(*act*) be a set of actual objects classified by domain experts. When the number of elements in *Set* is denoted by |*Set*|, the precision and recall are defined as follows:

$$precision = \frac{|Set(pred) \cap Set(act)|}{|Set(pred)|}, \quad recall = \frac{|Set(pred) \cap Set(act)|}{|Set(act)|} \tag{12}$$

The first experimental results for finding defects (Exp-1) are promising. Our system identified all defects effectively (Recall = 1.00, Precision = 0.95) from test images efficiently (Average response time = 0.64 s to process a single test image). It means that our system does not miss any defects while it allows a few (eight) false hits. The false hits were found because some defects identified turned out to be noises that had been generated from the image acquisition process, which is usual in the film-production industry.

The second experiment is to determine defect types (Exp-2), and Table 4 shows a confusion matrix of our empirical study results. As we can see from the table, most of defects are correctly classified. The recall is in the range 0.85–1.0 and on the average 0.95 while the precision is in the range 0.87–1.0 and on the average 0.96, as we observe in Fig. 16. Meanwhile, the execution time for classifying defect types (excluding the time for finding defects) is 0.23–0.45 s (Average response time = 0.32 s) to process a single test image. These results also come up to our expectations and we believe our method is applicable in related industries.

The third experiment is to show that our system is robust with respect to the rotational transformation for both finding defects and determining defect types (Exp-3). First, our system was also able to find all defect without any false dismissal (Recall = 1.00). It showed the same precision rate (Precision = 0.95) as the case with no transformation. It also performed efficiently (Average response time = 0.65 s), showing the similar performance. Next, an experimental result for determining defect types is shown in Fig. 17. We observe that the recall is in the range 0.60–1.0 and on the average 0.89 while the precision is in the range 0.86–1.0 and on the average 0.95. The effectiveness degrades for the rotational transformation, but not much. Thus, our system is regarded as 'robust' and is quite usable in the practical environment. For cunic defects, however, the recall is relatively low (Recall = 0.60). It is because a number of fine defects are distributed dispersedly in directions and thus some of them are considered as noises and eliminated. As future research, we plan to define features for such fine defects and devise an elaborate method to determine defect types based on the features.

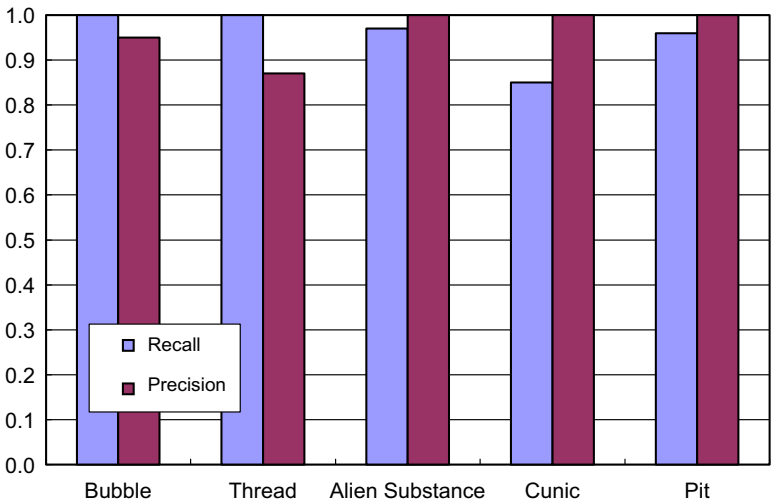

Fig. 16. Recall and precision for determining defect types.

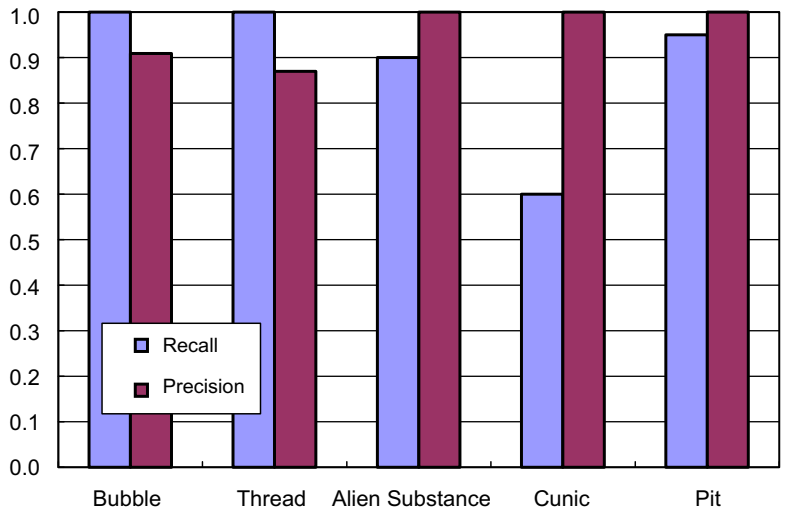

Fig. 17. Recall and precision for determining defect types for rotationally transformed defects.

5. Conclusions

In this paper we design and implement an effective defect inspection system that detects film defects and determines their types, and used in the last stage of manufacturing polarized films of TFT-LCD. The proposed system utilizes an image tiling technique to eliminate unnecessary work such as image filtering and blurring needed in the image pre-processing stage, and speed up in extracting defect regions from images. We use four features in determining defect types: the area ratio of ADR with respect to ADR's MBR, the relative center of ADR with respect to ADR's MBR, the contrast, and the ASM using texture features. Using those features we were able to determine defect types effectively.

To evaluate our proposed method, we have conducted three classes of experiments using the implemented system. First, our system identified all defects effectively and efficiently. It does not miss any defects while allowing a few false hits. Next, our system determined defect types properly, showing that the recall and precision are on the average 0.95 and 0.96, respectively. We believe our method is applicable in the related production field. Finally, our system is turned out to be robust with respect to the rotational transformation in finding defects and determining defect types.

An application that is emphasized in this paper is the inspection of film defects, but we believe other potential application areas can also benefit from the research. As the future work, we plan to study on applying the proposed method to the specialized application domains considering their own characteristics, such as topographical search and object recognition. We also plan to investigate the defect determination for cunic type which has shown a low recall rate, and the image segmentation method using pattern recognition techniques to improve the effectiveness and efficiency of our system.

References

Chang, N. S., & Fu, N. S. (1980). Query-by pictorial example. *IEEE Transactions On Software Engineering*.

Chang, Y. L., & Li, X. (1994). Adaptive image region growing. *IEEE Transactions on Image Processing, 3*, 868–872.

Cheng, H. D. et al. (2001). Color image segmentation: Advances and prospects. *Pattern Recognition, 34*, 2259–2281.

Choi, Y. (2003). Scene text extraction in natural images using hierarchical feature combining and verification. In *The 2nd KAIST-Tsinghwa Joint Workshop on Pattern Recognition*, pp. 76–102.

Chu, C., & Aggarwal, J. K. (1993). The integration of image segmentation maps using region and edge information. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*, 1241–1252.

Fan, J., & Yau, D. K. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing, 10*, 1454–1466.

Fan, J., Zeng, G., Body, M., & Hacid, M. S. (2005). Seeded region growing: An extensive and comparative study. *Pattern recognition letters, 26*(8), 1139–1156.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics. SMC, 3*(6), 610–620.

Hijjatoleslami, S. A., & Kittler, J. (1998). Region growing: A new approach. *IEEE Transactions on Image Processing, 7*, 1079–1084.

Jain, A. K., & Yu, B. (1988). Automatic text location in images and video frames. *Pattern Recognition, 31*(12), 2055–2076.

Jeong, K. W., Park, J. S., & Gang, C. G. (2003). A study on the defect inspection of the LCD polarized film using the vision system. *Journal of Industrial Science and Technology Institute, 17*(1), 47–53.

Jianfeng, L. (2005). Adaptive region growing algorithm in medical images segmentation. *Journal of Computer Aided Design and Computer Graphics, 17*(10), 2168–2173.

Kim, D. Y., Chung, S. M., & Park, J. W. (2006). Automatic navigation path generation based on two-phase adaptive region-growing algorithm for virtual angioscopy. *Medical Engineering and Physics, 28*(4), 339–347.

Kim, J. S., Kim, S. H., & Choi, H. W. (2005). Text region extraction of natural scene images using gray-level information and split/merge method. *Journal of Korea Information Science: Software and Applications, 32*(6), 502–511.

Lu, C. J., & Tsai, D. M. (2004). Defect inspection of patterned thin film transistor-liquid crystal display panels using a fast sub-image-based singular value decomposition. *International Journal of Production Research, 42*(20), 4331–4351.

Nakashima, K. (1994). Hybrid inspection system for LCD color filter panels. In *Proceedings of the 10th International Conference on Instrumentation and measurement Technology*, pp. 689–692.

Passat, N., Ronse, C., Baruthio, J., Armspach, J., Maillot, C., & Jahn, C. (2005). Region-growing segmentation of brain vessels: An atlas-based automatic approach. *Journal of Magnetic Resonance Imaging : JMRI, 21*(6), 715–725.

Pavlidis, T., & Liow, Y. T. (1990). Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*, 225–233.

Shih, F. Y., & Cheng, S. (2005). Automatic seeded region growing for color image segmentation. *Image and Vision Computing, 23*(10), 877–886.

Sokolov, S. M., Treskunov, A. S. (1992). Automatic vision system for final test of liquid crystal display. In: *Proceedings of the 1992 IEEE international conference on robotics and automation*, pp. 1578–1582.

USC-SIPI Image Database, http://sipi.usc.edu/services/database/.