

Real-time Defect Detection Method for Printed Images Based on Grayscale and Gradient Differences

Wang Yangping^{1,3,*}, Xu Shaowei¹, Zhu Zhengping², Sun Yue⁴ and Zhang Zhenghai⁵

¹School of Electronic & Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

²School of Electronic & Information Engineering, Lanzhou City University, Lanzhou 730070, China

³Gansu Provincial Engineering Research Center for Artificial Intelligence and Graphic & Image Processing, Lanzhou 730070, China

⁴Department of International Development, Community, and Environment, Clark University, Worcester, MA 01610, United States

⁵Lanzhou Bocai Technology Co. Ltd., Lanzhou 730060, China

Received 7 August 2017; Accepted 22 January 2018

Abstract

In a real-time quality inspection of printed matter based on machine vision, artifacts are induced by commonly used image difference methods, making the identification of defects difficult. Thus, to eliminate artifacts and improve detection rate of printing defects, this study proposed a method that combines grayscale and gradient differences. First, the grayscale difference between template image and inspected image was performed to determine the defect in the non-edge region according to the grayscale difference threshold of non-weighted neighborhood. Then, the gradient difference between the template image and inspected image was employed to determine the edge defect according to the grayscale difference threshold of weighted neighborhood. Finally, the difference artifacts were effectively eliminated by the two different image fusions and the real defects were retained. Experiments were conducted to compare the defect detection rate of printed image by using the traditional and proposed methods. Results demonstrate that for the most common dot defects the detection rate of the proposed methods is significantly higher than that of the traditional difference method due to the effective elimination of artifacts. The parallel acceleration based on compute unified device architecture (CUDA) enables the algorithm to speed up the defect detection of large print images by more than 60 times. The study provides significantly references for industrial inspection based on machine vision.

Keywords: Machine vision, Printed matters, Defect detection, Image difference, CUDA

1. Introduction

Modern printing industry is developing rapidly and the printing process tends to be high-speed, continuous, and geared toward mass production. However, printing defects may occur because of low precision of printing equipment, mismatch between equipment and image parameters, and equipment or material fault during image reproduction. Such defects cause inconsistent visual effect of the printed matter, thereby negatively affecting the quality of the product. Traditional printing defects are detected manually. However, the artificial detection is difficult to meet the requirement of batch production of packaging products because of high cost, slow speed and low recognition rate.. With the development of intelligent and informational technology, machine vision has been widely used in industrial inspection [1-5]. Compared with the traditional manual detection method, automatic detection based on machine vision can reduce the labor intensity of inspectors and improve detection efficiency and accuracy [6-7] with the feature of non-contact, fast speed, and anti-interference.

In the production process of printed matters, the detection system based on machine vision with the linear charge-coupled device camera can reliably record the high

definition real-time image of printed matter. Then, the quality of the printed matter is checked and the equipment is adjusted to prevent batch waste or pick out accidental defective products. Researchers have conducted a large number of studies on the defect detection of printed matter based on machine vision [8-12], and great research progress has been made on the online detection technology for printing defects. However, the detection of dirty spots, fog points, flying-ink, scraping and so on, which requires high detection accuracy, is seldom considered in the algorithm, and then missing or false inspection of such defects could be caused. The focus and difficulty of the research on the printing defects detection is how to improve the detection speed and the detection accuracy.

Based on the above analysis, the study presents a real-time defect detection algorithm for printed images based on image gradient and grayscale difference method. The image is divided into the edge and non-edge areas, and the different difference methods are used to determine the defect. The two difference results are fused to eliminate the difference artifact and improve the detection precision. The algorithm is accelerated in parallel to improve real-time performance based on compute unified device architecture (CUDA) [13].

2. State of the art

Scholars all over the world have conducted plenty of studies on automatic printed defect detection based on machine

*E-mail address: w_yp73@163.com

ISSN: 1791-2377 © 2018 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.

doi:10.25103/jestr.111.22

vision and proposed different methods. Tanimizutnj et al. [14] proposed index space method to detect printing defects. The algorithm uses the template image and the inspected image as the input images, and determines whether the printed matter is qualified by comparing the gray value of the corresponding pixel points. The images are operated pixel by pixel, which makes real-time poor. On the basis of above index space method, Mehenni et al.[15]proposed the n-tuple pixel matching method to simplify the complexity of the algorithm. Although the algorithm effectively improves the speed of detection, it is not strong enough to judge most types of printing defects. An automatic defect detection method that pays attention not only to shape-defects but also to color-defects is presented for the printed matters [16], which achieves better detection precision based on multi-feature. Luo et al.[17] developed an automatic detection algorithm based on artificial neural network. The method is able to inspect the defects of complex color prints under varying illumination conditions. Zhang and others proposed a printing defect detection algorithm based on template matching. To some extent, this method solves the drawbacks of manual detection, but is more suitable. for off-line detection system with poor real-time [18].In order to further improve the running speed of the detection algorithm, Wang et al. presented an image defect recognition algorithm based on dynamic threshold and hierarchical detection. By combining the dynamic threshold with the hierarchical detection, the detection of the whole printed image can be completed quickly. However, undetected phenomenon often exists[19].Yang developed a multi-template defect detection algorithm for printed product, which can effectively remove the contour artifacts in the subtraction images, retain the real printing defects and have strong robustness[20]. Aiming at the difference between the collected images and the standard images in the traditional food packaging printing defect detection system, Yang et al. proposed a method to detect the defects of packaging printed matters based on image registration. The method improves the reliability and stability of the system [21].

The research on the defect detection of printed matter based on machine vision has made great progress, whereas the online defect detection of printed matter requires high precision and real-time performance. For detecting printed defect, the image difference algorithm based on template matching is simple and intuitive, easy to be implemented. However, since the reference image and the inspected image are not acquired at the same time, and the production environment may be greatly changed, the relative coordinates of the camera and the subject are not the same world coordinates during the process of capturing image. Although some of the above problems can be solved by image pre-processing, image correction, image registration, and so on, the artifact contours and defects brought by the simple image difference method become indistinguishable, which reduces the accuracy of defect detection[22,23].To cope with the artifacts caused by image difference, various false-contour removal methods have been developed based on mathematical morphology (MM) [24,25],shading template[5,26], and neighborhood iterative difference [22]. In the MM-based method, the size of the structural element must be greater than those of false contours and false defects. However, when false defects are filtered in the printing process, small true defects are filtered as well, thus significantly decreasing the detection accuracy. Although the shading-template-based method is efficient in removing false contours in non-edge areas, false defects in the edge

area beyond the threshold value cannot be detected. The neighborhood iterative difference method can be used to significantly reduce false contours. However, iterative difference on the neighborhood reduces the real area of the defect, thereby affecting the accuracy of defect detection.

In the study, a template-matching-based difference algorithm is developed to perform online defect detection of printed matter produced at high speeds. To reduce the influence of artifact on the accuracy of defect detection, the inspected images are divided into edge and non-edge areas. Grayscale and gradient differences are combined to reduce false contours in the edge area while protecting the defective image as much as possible. A quadratic image difference method is used to eliminate the edge artifacts in the non-edge area, compensating for the sharp decrease in the detection accuracy due to the blurring of false contours and true defects. The proposed algorithm not only improves the computational accuracy but also shows the advantages of template matching methods, such as low complexity. Based on CUDA, parallel acceleration is performed to improve the efficiency and real-time performance of the algorithm.

The remainder of this study is organized as follows. Section 3 describes the proposed defect detection method for printed matters and parallel acceleration algorithm based on CUDA. Section 4 discusses the applicability of the method through case studies. Section 5 summarizes the conclusions.

3. Methodology

3.1 Defect detection algorithm based on image difference

For the defect detection method based on image difference, different parts of two images are extracted by subtracting the inspected image from a standard image (also referred to as the reference image) and by jointly using the judgment threshold. The size and shape of the connection area are then analyzed to determine whether the inspected image is defective. The traditional difference algorithm (TDA) is defined as follows:

$$f_{absTD}(x,y) = \begin{cases} 255, & |T(x,y) - D(x,y)| \geq \sigma \\ 0, & |T(x,y) - D(x,y)| < \sigma \end{cases} \quad (1)$$

where $T(x,y)$ represents a reference image, $D(x,y)$ represents the inspected image, and σ is the fixed threshold value. A point is defective when $f_{absTD}(x,y)=255$ and non-defective when $f_{absTD}(x,y) = 0$.

The reference image and the inspected image are not acquired simultaneously, and the production environment varies significantly. The coordinates of the camera and the photographed subject are different during the actual acquisition of the image, and their relative coordinates may change slightly. Although some of these problems can be solved using various methods, such as image pre-processing, image correction, and image registration, it is difficult to distinguish artifacts and defects using the simple image difference method.

The main cause of the artifact is the shock of equipment, change of light, and other factors. When the absolute difference image $f_{absTD}(x,y)$ is obtained using the reference image and the inspected image, the expansion telescopic image difference $T(x,y)$, noise image difference $f_{absTD}^c(x,y)$, and true defect image

difference $f_{absTD}^T(x, y)$ are included. The image difference due to the telescopic distortion $f_{absTD}^f(x, y)$ is the main part of the artifacts, whereas the noise difference $f_{absTD}^c(x, y)$ due to the change in the illumination and equipment transmission interference is the minor part of the artifacts. The noise image difference $f_{absTD}^c(x, y)$ and the true defect image difference $f_{absTD}^r(x, y)$ are widely distributed in the image, existing either in the edge area or in the non-edge area of the image with equal probability in theory. Telescopic distortion significantly affects the error between the template image and the inspected image in the edge area because of the degree of telescopic distortion, whereas its effect on the non-edge area of the image is negligible. Therefore, the telescopic distortion image difference $f_{absTD}^f(x, y)$ is in the vicinity of the edge of the image.

Fig.1(a) shows the reference image and Fig.1 (b) shows the inspected image with white spots and flying ink. The maximum and minimum defect sizes are $15 \times 8px$ and $2 \times 1px$, respectively. The minimum defect discrimination (gray-level difference between defect and background) is 13 gray levels, whereas the maximum defect discrimination (the gray levels of the defects and the background of the defects) is 150 gray levels. The defects are distributed throughout the edge and non-edge areas. Fig.1(c) shows the difference image of the absolute value difference between the images shown in Fig.1(a) and (b). The vast majority of defects can be directly resolved by the naked eye, but there are a large number of artifact contour noises. Fig.1(d) shows the image of Fig.1(c) after binarization using threshold 10. The defects of minimum discrimination can be clearly found, but a large number of artifacts are highlighted, which is detrimental for extracting the defect in the non-edge area, whereas the edge area is also submerged. The subsequent processing will be difficult.

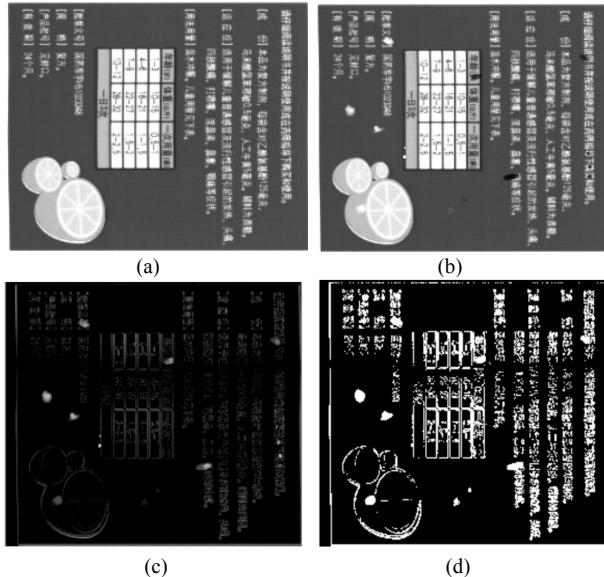


Fig.1. Simple image difference and image binarization.
(a)Reference image. (b) Inspected image. (c) Absolute image difference between the reference image and the inspected image. (d) Binarization image of Fig.1 (c) using threshold 10

3.2 Defect detection of printed matters based on image grayscale and gradient differences

The study proposes a defect detection method for printed matters based on gradient and grayscale differences. The method divides inspected images into edge and non-edge areas, and the defects in the edge and non-edge areas can be

identified using different methods by comparing with the reference image. Then, the defects in the edge and non-edge areas are extracted. Finally, the two detection results are combined to improve the detection accuracy. Fig.2 shows the proposed algorithm.

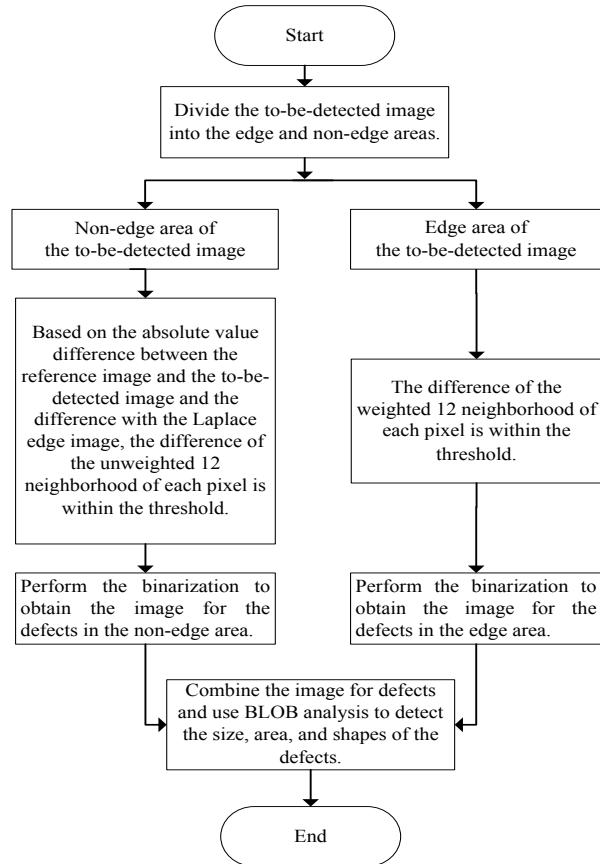


Fig.2. Flowchart of the method

The absolute image difference $f_{absTD}(x, y)$ can be calculated using the reference image $T(x, y)$ and the inspected image $D(x, y)$ as follows:

$$f_{absTD}(x, y) = |T(x, y) - D(x, y)| \quad (2)$$

where $f_{absTD}(x, y)$ includes telescopic distortion difference $f_{absTD}^f(x, y)$, noise difference $f_{absTD}^c(x, y)$, and true defect $f_{absTD}^r(x, y)$.

Take the Laplace edge image $f_{Laplace}(x, y)$ of the reference image $T(x, y)$, and perform the binarization on $f_{Laplace}(x, y)$ with a threshold of 10 to filter the influence of the non-edge area. $f_{Laplace}(x, y)$ is expanded and enhanced to obtain $f_{Laplace}^E(x, y)$ by the size of the morphological expansion kernel estimated on the basis of the stability of the device, which can be considered as a range mask in the edge area of the image.

Assuming that the image difference in the non-edge area is $f_{NEdge}(x, y) = 2 * [f_{absTD}(x, y) - f_{Laplace}(x, y)]$, the binary image of the defects in the non-edge area $f_{SUB}^{NEdge}(x, y)$ can be taken by performing binarization operation with a threshold of 10 to filter the noise and highlight the inspected object. As shown in Fig.3, the objects to be detected are defects in the non-edge area, and a low grayscale threshold (at least

greater than 10) and various types of small-sized defects in the non-edge area can be detected.

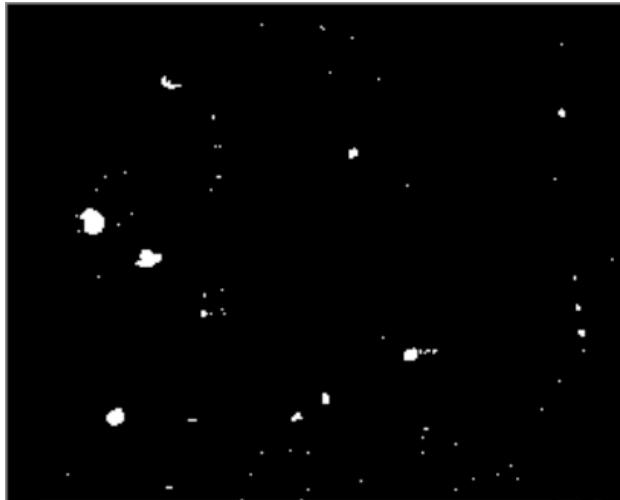


Fig. 3. Preliminary detection results of the defects in the non-edge area

The edge image $f_{Edge}^T(x, y)$ of the reference image $T(x, y)$ and the edge image $f_{Edge}^D(x, y)$ of the inspected image $D(x, y)$ are obtained. When the pixel in the absolute value difference image $f_{absID}(x, y)$ is greater than the threshold value of 10, it is a defect or an artifact. Under this condition, whether it is in the edge area or not needs to be identified on the basis of the edge image $f_{Edge}^T(x, y)$ of the reference image. For the edge area, whether the difference of the weighted 12 neighborhood of each pixel point between the reference image $T(x, y)$ and the inspected image $D(x, y)$ is within the range of the threshold A and whether the pixel difference between $f_{Edge}^T(x, y)$ and $f_{Edge}^D(x, y)$ is within the threshold B should be checked. If the difference presented above is within the thresholds, then the point is an artifact contour, otherwise, it is a true defect point. This operation will make the edge defects detectable. However, detecting defects of small sizes or those that have a low grayscale level is difficult. For the non-edge area, whether the difference of the weighted 12 neighborhood of each pixel point between the reference image $T(x, y)$ and the inspected image $D(x, y)$ is within the range of the threshold C should be checked. If the difference is within the threshold, then the point is an artifact contour, otherwise, it is a true defect point. This operation will make defects that have a low grayscale level and small-sized true defects in the non-edge area undetectable. However, noise can be filtered and a part of the missing defects in the previous operation in the non-edge area can be detected. The defects in the edge area and the binary image $f_{SLB}^Edge(x, y)$ for a part of the defects in the non-edge area can be obtained after identifying the true defect points, as shown in Fig.4. The objects to be detected include the defects in the edge area and a part of the missing defects in the previous operation in the non-edge area. Nevertheless, defects with low grayscale level and too small sizes are undetectable.

The binary images for the defects in the non-edge area $f_{SLB}^{NEdge}(x, y)$ and the defects in the edge area $f_{SLB}^{Edge}(x, y)$ are combined to obtain $f_{SLB}(x, y)$. As shown in Fig.5, the yellow-colored region indicates the binary image for the detected defects in the non-edge area $f_{SLB}^{Edge}(x, y)$, the cyan region indicates the binary image for the detected defects in the

edge area, and the red regions indicates the overlapped results of the two methods.



Fig. 4. Initial detection results for the defects in the edge area

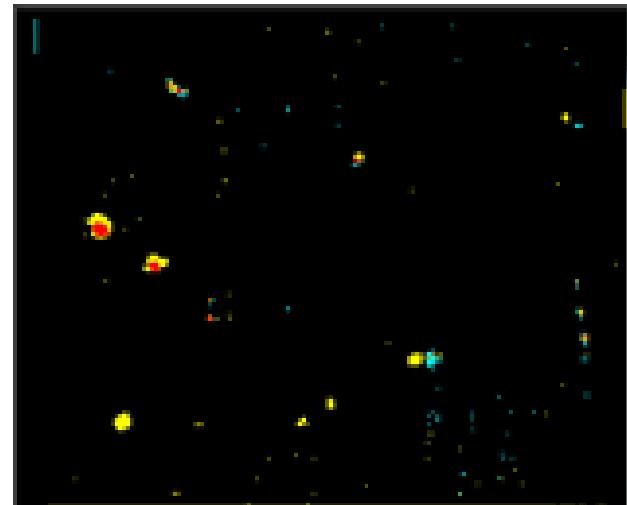


Fig. 5. Preliminary detection results after combining the defects

The blob analysis method [27] is used on $f_{SLB}(x, y)$ to calibrate all kinds of shape defects with area greater than $2px$, and marks are drawn, as shown in Fig.6. A defect on the edge of the lower right part is missed, and the noise in the middle of the image is negligible.

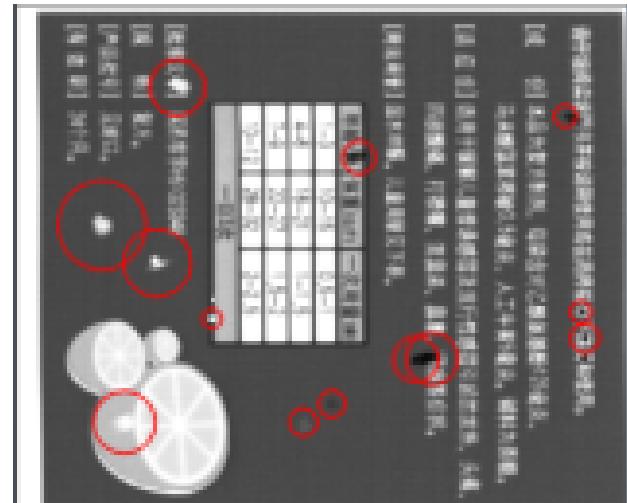


Fig. 6 Final detection results after blob analysis

3.3 Parallel acceleration of proposed algorithm based on OpenCV and CUDA

Machine vision online detection system requires high real-time algorithm. In this proposed method, image edge processing, image morphological processing, threshold segmentation, and absolute value difference operation of image are suitable for parallel processing of image pixels. In this study, the GPU module of OpenCV and CUDA API were used to speed up the image processing based on CUDA. The main functions in the OpenCV GPU module are given as follows:

```

M_imgT_gpu.upload (m_imgT); // Use OpenCV GPU
    module to upload image data to GPU
M_imgD_gpu.upload (m_imgD);
Cv :: gpu :: Laplacian (m_imgT_gpu, m_imgTsobel_gpu,
    m_imgT.depth (), 1, 1); // Use GPU to split the image threshold
Cv :: gpu :: threshold (m_imgTsobel_gpu, m_imgTedge_
    subEdge_gpu, _threshold, 255, THRESH_BINARY); // Use GPU to dilate the
    image
Cv :: gpu :: dilate (m_imgTsobel_gpu, m_imgTedge_
    subEdge_gpu, element); // Use the GPU to dilate the
    image
Cv :: gpu :: absdiff (m_imgT_gpu, m_imgD_gpu,
    m_imgOldMethodDiffout_gpu); // Use GPU to
    perform the absolute value image difference
Cv :: gpu :: morphologyEx (m_imgNewSubDiffOut_
    subEdge_gpu, imgNewSubDiffOut_subEdge_gpu,
    MORPH_OPEN, element); // Use GPU to implement
    the morphological opening of the image

```

A kernel function is customized based on CUDA API to accelerate the computation. In the kernel function, the dimensions of the block and grid need to be set for the algorithm and hardware parameters. Threads in the same block can share data through shared memory to reduce the number of data reads from the graphics card. With a large block dimension, more threads participate in the computation process, but it will consume more registers and shared memory, thereby reducing the number of active thread blocks (active block) on streaming multiprocessor. In this study, CUDA Occupancy calculator tool in CUDA SDK is used to calculate the occupancy of the CPU computational unit with different block dimensions, as shown in Fig.6, where block size 512 is one of the optimal solutions.

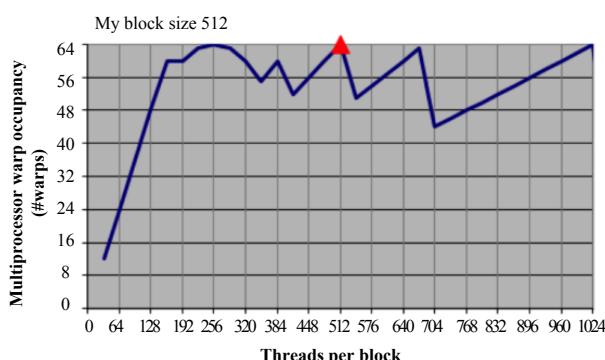


Fig. 6. Occupancy rate for the computational unit of GPU

Once the dimensions of the block are determined, the dimensions of the grid can be calculated using the following equation.

$$\begin{cases} GRID_X = \frac{WIDTH + BLOCK_X - 2}{BLOCK_X - 2} \\ GRID_Y = \frac{HEIGHT + BLOCK_Y - 2}{BLOCK_Y - 2} \end{cases} \quad (3)$$

where GRID_X and GRID_Y are the numbers of grids on the X and Y axes, respectively. WIDTH and HEIGHT are the width and height of the image to be processed. BLOCK_X and BLOCK_Y are the numbers of blocks on the X and Y axes. When the optimal solution for the block size is used, the number of registers used by each thread and the size of the shared memory used by each block should be configured. As shown in Fig.7 and Fig.8, one of the optimal solutions is that the number of registers for each thread is 8 and the shared memory requires 4,096 bytes. Considering the above parameters, the number of active threads on each GPU reaches 2,048, and the occupancy rate for each computational unit on the GPU reaches 100%.

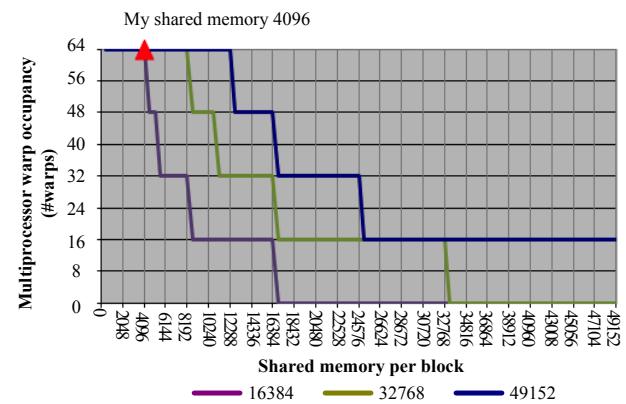


Fig. 7. Occupancy rate of the shared memory of GPU

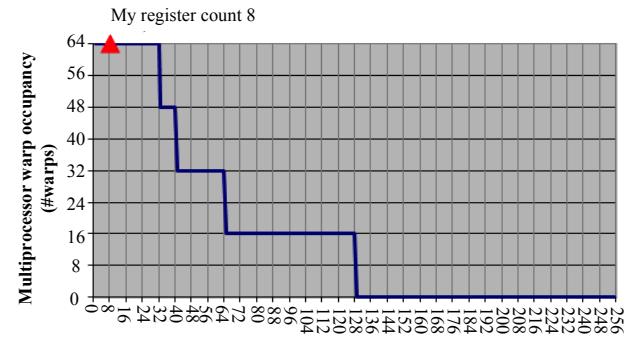


Fig. 8. Occupancy rate of the registers of GPU

In this study, the parallelism reduction strategy is used in the parallel processing. The location and computing relationship between each pixel point and the neighborhood pixels is handled by a GPU thread. In the custom kernel function, the pixel index of the inspected image can be obtained using the kernel function, and the corresponding pixel indices for the neighboring area and the template image can be computed. The area for each pixel is identified on the basis of the pixel index, and the identification results are assigned to different flows. The specific code is given as follows.

```

int x = threadIdx.x + blockIdx.x * blockDim.x; // column
int y = threadIdx.y + blockIdx.y * blockDim.y; // row

```

For the image in the non-edge area, whether the 12 neighborhood mean difference is within the threshold for each pixel should be calculated, based on which we can directly determine whether the pixel point is a false defect or a true one. For the image in the edge area, in addition to the aforementioned check, the value of the gradient difference should be determined. When the two conditions are satisfied, the pixel point can be identified as either a false or a true defect. The algorithm is as follows:

```

if (data_Tedge(y,x).x<=10)//The point is in the non-edge
area
{
    if (abs(adjacentAVE (data_T,x,y,ADJACENT_
KERNEL_12)-adjacentAVE(data_D,x,y,
ADJACENT_KERNEL_12))<TH_A)
    {
        data_TH_result(y,x).x = 0; //false defects
    }
    else
    {
        data_TH_result(y,x).x =255;//true defects
    }
}
else// The point is in the edge area
{
    if(abs(adjacentAVE(data_T,x,y,ADJACENT_K
ERNEL_12,TURE)-adjacentAVE
(data_D,x,y,ADJACENT_KERNEL_12,
TURE))<TH_A&&abs(data_Tedge(y,x).x-
data_Dedge(y,x).x)< TH_B)
    {
        data_TH_result(y,x).x=0; //false defects
    }
    else
    {
        data_TH_result(y,x).x=255;//true defects
    }
}

```

4. Result analysis and discussion

In the experiment, the printed images were acquired using the three-line array color camera PC-30-04K80 made by DALSA, the series lens FV5026W-F2.6 made by Utron, and the LED high-frequency strip light source OPT-LSG806-W. A computer with an Intel Core i5-2310 CPU and 4 GB memory and NVIDIA Tesla k20 (2496 CUDA Cores) were used to detect defects in the printed images via Visual Studio 2010 and OpenCV.

A large number of printed images were tested for line defect and dot defect detection. Fig.9–Fig.11 show two sets of experimental data. Fig.9 shows a qualified print as a reference image for defect detection.

Fig.10 shows printed images with line defects corresponding to the reference images in Fig.9. In Fig.10 (a), 25 scratches or ink bar defects are found, the size of the line defects is large, and the minimum defect size is $2 * 35$ (width * length). 25 scratches or ink bar defects are found in Fig.10 (b), including various size line defects with a minimum defect size of $2 * 3$. These experiments compared the results of the proposed difference algorithm (PDA) and the traditional difference algorithm (TDA) for the line defect

detection in Fig.10 as shown in Table 1. PDA can reliably detect all the defects in two sets of defect images, while TDA has a larger false detection rate when the defect size is smaller.



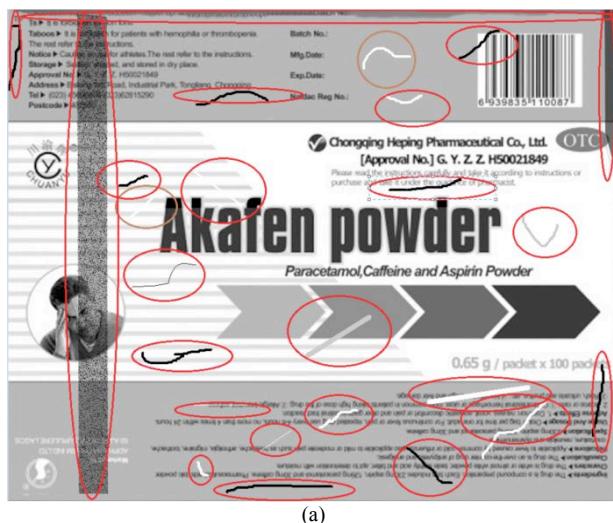
Fig.9. Reference images.(a)Reference image A. (b) Reference image B

Fig.11 shows printed images with dot defects corresponding to the reference image in Fig.9. In Fig.11(a1) and Fig.11(b1), 9 white spots or flying ink defects are found. The size of the dot defects is large and the minimum defect size is $6 * 6$. 20 and 16 white spots or flying ink defects are found in Fig.11(a2) and Fig.11(b2), respectively, which include various size dot defects with a minimum defect size of $1 * 1$. The experiment results were compared those of the PDA and TDA for the dot defect detection in Fig.11, as shown in Table 2. When defects are large, as shown in Fig.11(a1) and Fig.11(b1), two of the algorithms can find all defects. However, when small defects are present, the recognition error rate appears in both algorithms. PDA can reliably detect the edge and non-edge defects of the printed image if the defects are more than 6 pixels, but when the defects are less than 4 pixels, the algorithm is obviously missing, especially in the edge area. TDA will have a high

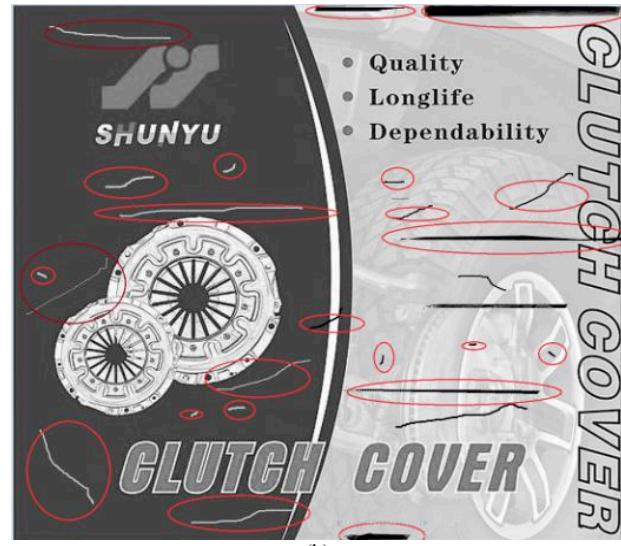
detection error rate when the defect size is small or defect gray contrast is not obvious.

Table 1. Comparison of line defect detection

Items	Line defect image A	Line defect image B
Defect type	Scratches, ink bars	Scratches, ink bars
Minimum defect size (pixel)	2 × 35	2 × 3
Minimum grayscale difference of defects	75	47
Actual defects	25	25
Defects detected	PDA TDA	25 25 19
Defect detection rate (%)	PDA TDA	100% 100% 76%
Missed rate (%)	PDA TDA	0% 0% 24%



(a)



(b)

Fig.10. Inspected images with line defects.

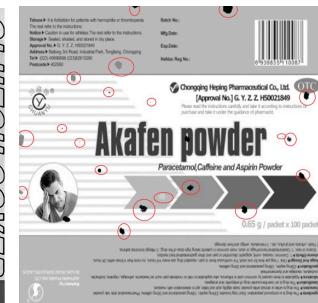
(a) Image A with line defects. (b) Image B with line defects



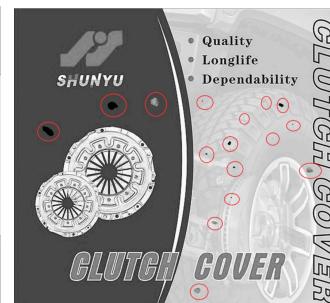
(a)



(b1)



(a2)



(b2)

Fig.11. Inspected images with dot defects.

(a1) Image A1 with dot defects. (b1) Image B1 with dot defects. (a2) Image A2 with dot defects. (b2) Image B2 with dot defects

Table 2. Comparison of dot defect detection

Items	Dot defects Image A1	Dot defects Image B1	Dot defects Image A2	Dot defects Image A2	Dot defects Image B2	Dot defects Image B2
Defect type	White spot, fly ink					
Minimum defect size (pixel)	6 × 6	6 × 6	1 × 1	2 × 3	1 × 1	2 × 2
Minimum grayscale difference of defects	20	15	20	20	13	13
Actual defects	9	9	20	16	16	13
Detected defects	PDA TDA	9 9	16 14	16 14	12 10	12 10
Defect detection rate (%)	PDA TDA	100% 100%	100% 100%	80% 70%	100% 87.5%	75% 62.5%
Missed rate (%)	PDA TDA	0% 0%	20% 30%	0% 12.5%	25% 37.5%	7.7% 23.1%

The average running time was obtained by conducting 10 tests on the images with different resolutions using the

CUDA (CPU + GPU) parallelism acceleration for the proposed method. Compared to the method without CUDA

(CPU only), the computation efficiency was improved significantly using the CUDA, as listed in Table 3.

Table. 3. Speedup of the proposed algorithm based on CUDA

Resolution	Running time (ms)		Acceleration ratio
	CPU	CUDA (CPU + GPU)	
128 × 128	16.954	2.002	8.5
256 × 256	61.036	3.764	16
512 × 512	256.351	8.658	30
1024 × 1024	1048.476	19.648	53
2048 × 2048	4078.571	64.764	63
4096 × 4096	16926.071	264.237	64

5. Conclusions

When checking the appearance defect of print matters in real time on the production line, the common method of direct difference comparison between real-time image and reference template image would reduce defect detection rate with generated artifacts. The study proposed a method that detected the defects of printed matters by taking the grayscale and gradient difference on the edge and non-edge areas between reference image and inspected image, respectively, and then fusing their results, so as to eliminate artifacts and improve the detection rate of defects. The following conclusions could be drawn:

(1) Compared with the traditional difference method, the combination difference of grayscale and gradient maintains the original defect structure characteristic of the inspected printed matters and removes unrelated artifact contour. The proposed method overcome the phenomenon of neglecting small defects caused by the elimination of the artifact

contour and improves the accuracy of the difference method for extracting the appearance defects of the printed matter.

(2) The improved method and the traditional difference method are used to detect two types of defects of line and dot, respectively. The results show that the improved method can effectively eliminate the influence of artifacts, and the detection rate of the most frequent dot defect is significantly higher than that of direct difference method.

(3) The speedup of defect detection algorithm proposed in this study can achieve more than 60 times for large printed images based on CUDA, which improves significantly the defect detection in real-time performance.

The defect detection method of printed matters proposed in this study improved the defect detection rate, which is simple and easy to be implemented due to the features of difference algorithm. This method has a certain reference value for on-line real-time industrial product appearance detection based on machine vision. However, the algorithm is sensitive to the gray value of the image, which is easily affected by the illumination change. In application, the environmental light needs to be stable.

Acknowledgements

The work was supported by the Program for Changjiang Scholars and Innovative Research Team in Universities (Grant No. IRT_16R36), the Science and Technology Project of Gansu Colleges and Universities (Grant No. 2017D-08), and Lanzhou Talent Innovation and Entrepreneurship Project (Grant No. 2015-RC-8).

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence



References

1. Pacella, M., Grieco, A., Blaco, M., "Machine vision based quality control of free-form profiles in automatic cutting processes". *Computers & Industrial Engineering*, 109, 2017, pp.221-232.
2. Zhang, M., Wu, J.L., Lin, H.F., Yuan, P., Song, Y.A., "The application of one-class classifier based on CNN in image defect detection". *Procedia Computer Science*, 114, 2017, pp. 341-348.
3. Payman, M., Alireza, S., Hossein, P., "Computer vision-based apple grading for golden delicious apples based on surface features". *Information Processing in Agriculture*, 4(1), 2017, pp.33-40.
4. Jian, Ch.X., Gao, J., Ao, Y.H., "Automatic surface defect detection for mobile phone screen glass based on machine vision". *Applied Soft Computing*, 52, 2017, pp. 348-358.
5. Guan, Y. Y., Ye, Y. C., "Printing defects detection based on two-times difference image method". *Applied Mechanics & Materials*, 340, 2013, pp.512-516.
6. Vedang, C., Brian, S., "A comparative study of machine vision based methods for fault detection in an automated assembly machine". *Procedia Manufacturing*, 1, 2015, pp. 416-428.
7. Zhang, H., Shi, T., He, S. CH., "Defect detection system of medical infusion container combination cover based on reverse P-M diffusion", *Journal of Electronic Measurement and Instrument*, 29(5), 2015, pp.692-700.
8. VANS M. "Automatic visual inspection and defect detection on variable data prints". *Journal of Electronic Imaging*, 20(1), 2011, pp.1-13.
9. Wang, Y.W., Qu, G.T., Liu X.L., "Image subtraction detection algorithm for surface defect". *Journal of Computer-Aided Design & Computer Graphics*, 28(10), 2016, pp.1769-1704.
10. LUNDSTROM, J., VERIKAS, A., "Assessing print quality by machine in offset colour printing". *Knowledge-Based Systems*, 37, 2013, pp.70-79.
11. Shankar, N. G., Ravi, N., Zhong, Z. W., "A real-time print-defect detection system for web offset printing". *China Printing & Packaging Study*, 42(5), 2009, pp.645-652.
12. Luo, J., Zhang, Z., "Automatic color printing inspection by image processing". *Journal of Materials Technology*, 13(9), 2003, pp.373-378.
13. Ahamed, A.K.C., Magoulès, F., "Conjugate gradient method with graphics processing unit acceleration: CUDA vs OpenCL original research article". *Advances in Engineering Software*, 111, 2017, pp.32-42.
14. Tanimizu, K., Meguro, S., "High-speed defect detection method for color printed matter". In: *Industrial Electronics Society 16th Annual Conference of IEEE*, Pacific Grove , USA: IEEE, 1990, pp.635-658.
15. Mehenni, B., Wahab, M.A., "ARPIS:automatic pattern recognition and inspection system". In: *CompEuro '93, Computers in Design, Manufacturing, and Production, Proceeding*, Pris-Evry, France: IEEE, 1993, pp.23-28.
16. Ishimaru, I., Hata, S., Hirokar, M. "Color-defect classification for printed-matter visual inspection system". In: *Proceedings of the 4th World Congress on Intelligent Control and Automation*, Shanghai, China: IEEE, 2002, pp. 3261-3265.
17. Luo, J., Zhang, Z., "Automatic colour printing inspection by image processing". *Journal of Materials Processing Tech*, 139(1), 2003, pp.373-378.
18. Zhang, Y. J., Huang, X. Y., Li, R., "A preliminary scheme for automatic detection of fine presswork defect". *Chinese Journal of Stereology & Image Analysis*, 6(2), 2001, pp.109-113.
19. Wang, F., Ran, Q.Q., "Research and application of image defect recognition algorithm based on dynamic threshold and stored Inspect". *Journal of Northern Jiaotong University*, 2, 2002, pp.19-22.

20. Yang, O., Guo, B., Hu, T., Guang, X., "A real-time vision system for defect detection in printed matter and its key technologies". In: *2007 Second IEEE Conference on Industrial Electronics and Application*, Harbin, China: IEEE, 2007,pp.2157-2161.
21. Yang, Z.B., Dai, X.H., "Printing defects detection and realization in food packaging based on image registration". *Computer Science*, 42(8),2015,pp.319-322.
22. Yang, X.M.,Wu, S.H., "A rapid defect detecting algorithm for printed matter on the assembly line".In: *International Conference on Systems and Informatics*.Yantai,China:IEEE, 2012,pp.1842-1845.
23. Huang, X.Q., Luo, X.B., "A real-time algorithm for aluminum surface defect extraction on non-uniform image from CCD camera". In: *International Conference on Machine Learning and Cybernetics*, Lanzhou, China:IEEE, 2014,pp.556-561.
24. Lin, L.Z., Zhou, L.S., Wan, J.D., "Study of PCB automatic optical inspection system based on mathematical morphology". In: *International Conference on Computer Technology and Development*, Kota Kinabalu, Malaysia :IEEE, 2009,pp.405-408.
25. Ukhanova, L.N., Frankfurt, M.O., "Application of mathematical morphology in the inspection on defects of weld images with X ray". *Modern Welding Technology*, 47(6),2010,pp.1396-1400.
26. Ou, Y., Tao, H., Xuan, G., "An automation system for high-speed detection of printed matter and defect recognition".In: *International Conference on Integration Technology*,Shenzhen, China:IEEE, 2007 pp.213-217.
27. Chethan,Y.D., Krishne gowda,Y.T., Ravindra,H.V., Bharath Kumar, S., "Machine vision for tool status monitoring in turning inconel 718 using blob analysis". *Materials Today: Proceedings*, 2(4-5), 2015, pp.1841-1848.