Adaptive Directional Cubic Convolution for Integrated Circuit Chip Defect Image Interpolation

Yuan Chao[1], Chengxia Ma[1], Wentao Shan[1], Junping Feng[1], Zhisheng Zhang[2]

[1]School of Mechanical Engineering, Jiangsu University of Technology, Changzhou, 213001, Jiangsu, China

[2]School of Mechanical Engineering, Southeast University, Nanjing, 211189, Jiangsu, China

Abstract—**An adaptive directional cubic convolution interpolation method for integrated circuit (IC) chip defect images is proposed in this paper, to meet the challenge of preserving edge and texture information. In the proposed method, Otsu thresholding technique is employed to distinguish strong edge pixels from weak ones and texture regions, and estimate the direction of strong edges, adaptively. Boundary pixels are pre-interpolated using the original bicubic interpolation method to help improve the interpolation accuracy of the interior pixels. The experimental results of both classic test images and IC chip defect images demonstrate that the proposed method outperforms the competing methods with better edge and texture preservation, interpolation quality, more natural visual effect of the interpolated images and reasonable computational time. The proposed method can provide high quality IC chip images for defect detection and has been successfully applied on practical vision inspection for IC chips.**

Keywords—**image interpolation; integrated circuit chip; Otsu thresholding; bicubic**

I. INTRODUCTION

Automated vision inspection on integrated circuit (IC) chip packages plays an overwhelmingly important role during the final testing process, which ensures the quality and reliability of the packages and the circuits. The quad flat no-lead (QFN) package is a typically desirable IC chip package for high speed and high power components, benefiting from its electrical and thermal performance [1]. However, different types of defects may appear on the surface of QFN packages, which will certainly seriously affect the packages' stability and durability. Therefore, defect detection is performed to detect and further classify these defect products in vision inspection [2]. One important factor that affects the defect detection

accuracy is the resolution of the target images. High resolution (HR) defect images can provide more details of the defect regions and better perceptual quality for both machine vision and manual review. Since sensor and optics manufacturing technology are limited and demand more cost in IC chip packages vision inspection, image interpolation becomes feasible and promising by estimating unknown pixel values in the HR image, from known pixel values in its corresponding low resolution (LR) image [3].

Classical polynomial-based interpolation methods include bilinear [4], bicubic [5], and cubic spline interpolation [6], which are also known as non-adaptive image interpolation methods. These methods are preferable because of their computational simplicity. However, image pixels are treated indiscriminately, geometric regularities and discontinuities are disregarded in LR images in these methods. As a result, undesirable artifacts such as blurring, blocking and ringing around edges tend to occur frequently [7]. To remedy these disadvantages, several adaptive image interpolation methods have been proposed and applied to different image superresolution scenarios in recent years [8-16]. Missing pixels in HR images are estimated by using edge information in LR images, i.e., geometric regularity, which refers to the smoothness constraint along the edge orientation as opposed to the sharpness constraint across the edge orientation [14]. In [14] Li and Orchard proposed a new edge-directed interpolation (NEDI) method, in which edges orientation is estimated implicitly by exploiting their geometric duality property between the LR and HR covariance. Simulation results have proved that NEDI better preserves the geometric regularity and generates higher visual quality images compared with those of bilinear and bicubic, however, the burden of computational complexity still remains a tough task. Stankevich et al. [15]

proposed an subpixel-shifted satellite image superresolution method based on the correlation algorithms, statistical regularization, and filtering in frequency domain. The spatial resolution of large-size satellite images can be improved by a reasonable result of 48.7%. However, the processing efficiency of the proposed method still needs optimizing. Zhou et al. [16] proposed the directional cubic convolution (DCC) interpolation algorithm, in which the gradients of two orthogonal directions are compared to estimate local edge strength and direction in local window, and then the missing pixels are interpolated via cubic convolution interpolation along the estimated edge direction for strong edges. Missing pixels along weak edges or in texture regions are interpolated by combining the two orthogonal directional gradients and the specific parameters are determined experimentally. As a result, the interpolation quality varies in edge preserving in different images.

In this paper, we propose an adaptive directional cubic convolution (ADCC) interpolation method that employs Otsu thresholding to distinguish strong edge pixels from weak ones and texture regions, and estimate the direction of strong edges, adaptively. ADCC is then applied to perform image interpolation on IC chip, typically, QFN package defect images. The remainder of this paper is organized as follows. Section 2 provides an overview of DCC. Section 3 presents the ADCC. Section 4 presents the related experiments and results analysis. Finally some conclusion are made in section 5.

II. DIRECTIONAL CUBIC CONVOLUTION INTERPOLATION

As adopted in most of the methods, DCC algorithm firstly expands the original LR image I_1 with the size of $H \times W$ into an HR image I_h with the size of $(2H\text{-}1) \times (2W\text{-}1)$, for a scaling factor of 2, where $I_1(i,j) = I_h(2i-1, 2j-1)$, $1 \le i \le H$, $1 \le j \le W$. As shown in Fig.1 and Fig.2, the original pixels from the LR image are denoted in solid black circles, while the missing pixels are in white squares and circles. All the missing pixels in the HR image are interpolated in two stages. In stage 1, pixels in white circles with coordinates of $(2i, 2j)$ are interpolated as shown in Fig.1. Then in stage 2, pixels in squares can be interpolated based on pixels in solid black circles from the LR image and pixels in solid gray circles calculated in stage 1, as shown in Fig.2. Two types of missing pixels are interpolated in stage 2, denoted in gray squares with coordinates of $(2i, 2j+1)$, and black squares with coordinates of $(2i+1, 2j)$.

A. Interpolation stage 1

In stage 1, in a 7×7 neighborhood with the central coordinates of $I_h(2i, 2j)$ denoted in white squares, the two orthogonal directional gradients in the diagonal 45° and 135° are calculated as in Eq.(1) and Eq.(2), as shown in Fig.1(a) and (b), respectively.

$$G_{45°} = \sum_{m=3,\pm1} \sum_{n=3,\pm1} \left| I_h(2i+m, 2j-n) - I_h(2i+m-2, 2j-n+2) \right| \quad (1)$$

$$G_{135°} = \sum_{m=3,\pm1} \sum_{n=3,\pm1} \left| I_h(2i+m, 2j+n) - I_h(2i+m-2, 2j+n-2) \right| \quad (2)$$

Since the gradient along the edge is smaller, the edge direction can be estimated using the ratio of the above two orthogonal directional gradients as

$$\begin{cases} \text{if } \dfrac{1+G_{135°}}{1+G_{45°}} > T, \text{ the pixel } I_h(2i,2j) \text{ is on 45° strong edge} \\[2mm] \text{else if } \dfrac{1+G_{45°}}{1+G_{135°}} > T, \text{ the pixel } I_h(2i,2j) \text{ is on 135° strong edge} \\[2mm] \text{else the pixel } I_h(2i,2j) \text{ is on weak edge or in texture region} \\ \text{end} \end{cases} \quad (3)$$

where the threshold T is set to 1.15 based on Kodak color images [17] interpolation experiments.

Missing pixels in this stage can be calculated by the four known pixels along the detected edge and the cubic convolution kernel $f=[-1,9,9,-1]/16$ [5], as shown in Fig.1(c). The missing pixels p along the 45° diagonal direction are calculated as p_1 in Eq.(4) while 135° as p_2 in Eq.(5).

$$p_1 = f \cdot v_1^T$$
$$= f \cdot [I_h(2i-3, 2j+3), I_h(2i-1, 2j+1), I_h(2i+1, 2j-1), I_h(2i+3, 2j-3)]^T \quad (4)$$

$$p_2 = f \cdot v_2^T$$
$$= f \cdot [I_h(2i-3, 2j-3), I_h(2i-1, 2j-1), I_h(2i+1, 2j+1), I_h(2i+3, 2j+3)]^T \quad (5)$$

Non-strong-edge pixels p are calculated by combining p_1 and p_2 with weights w_1 and w_2, as in Eq.(6), where k is set to 5 experimentally to adjust the weights of the two interpolation coefficients.

$$\begin{cases} p = \dfrac{w_1 p_1 + w_2 p_2}{w_1 + w_2} \\[2mm] w_1 = \dfrac{1}{1+G_{45°}^k} \\[2mm] w_2 = \dfrac{1}{1+G_{135°}^k} \end{cases} \quad (6)$$

B. Interpolation stage 2

Pixels $I_h(2i, 2j)$ interpolated in stage 1 are considered as known pixels in this stage, denoted in solid gray circles, and are involved in the same calculation process of $I_h(2i, 2j+1)$ and $I_h(2i+1, 2j)$. In this stage, the edge detection and pixel interpolation are identical for both directions and similar to the previous stage. In a 5×5 neighborhood, the vertical and horizontal gradients are calculated as in Eq.(7) and Eq.(8), shown in Fig.2(a) and (b), respectively.

$$G_{90°} = \sum_{m=0,2} \sum_{n=\pm1} \left| I_h(2i-m, 2j+n) - I_h(2i-m+2, 2j+n) \right|$$
$$+ \sum_{n=0,\pm2} \left| I_h(2i-1, 2j+n) - I_h(2i+1, 2j+n) \right| \quad (7)$$

$$G_{0°} = \sum_{m=\pm1\,n=0,2}\sum |I_h(2i+m,2j-n) - I_h(2i+m,2j-n+2)|$$
$$+ \sum_{m=0,\pm2} |I_h(2i+m,2j-1) - I_h(2i+m,2j+1)| \quad (8)$$

Missing pixels in this stage can be calculated by the four known pixels along the detected edge as shown in Fig.2(c). The missing pixels p along the vertical direction are calculated as p_1 in Eq.(9) while horizontal as p_2 in Eq.(10).

For non-strong-edge pixels in this stage, the calculation is similar to Eq.(6) in the previous stage.

$$p_1 = f \cdot v_1^T$$
$$= f \cdot [I_h(2i-3,2j), I_h(2i-1,2j), I_h(2i+1,2j), I_h(2i+3,2j)]^T \quad (9)$$

$$p_2 = f \cdot v_2^T$$
$$= f \cdot [I_h(2i,2j-3), I_h(2i,2j-1), I_h(2i,2j+1), I_h(2i,2j+3)]^T \quad (10)$$
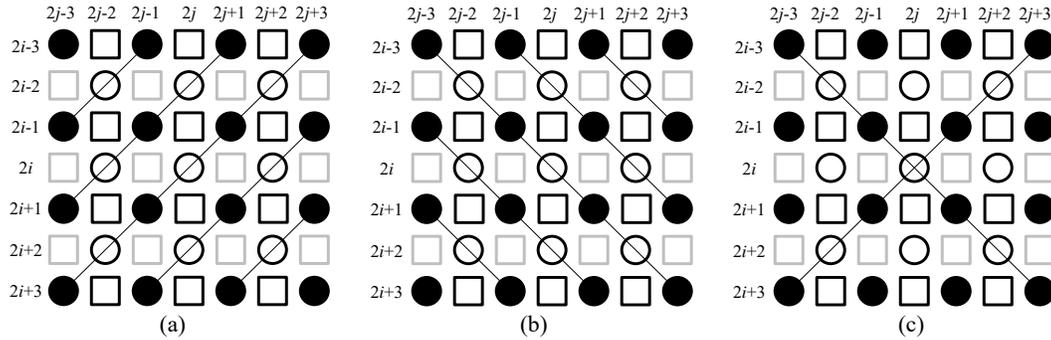
Fig.1. Edge direction determination and missing pixel value computation in stage 1 in DCC; a) gradients in 45° direction; b) gradients in 135° direction; and c) missing pixel calculation along 45° and 135° directions.
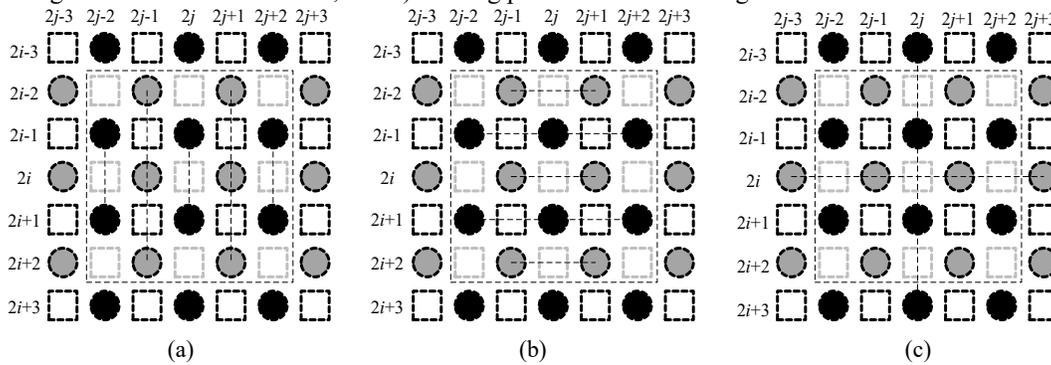
Fig.2. Edge direction determination and missing pixel value computation in stage 2 in DCC; a) gradients in vertical direction; b) gradients in horizontal direction; and c) missing pixel calculation along vertical and horizontal directions.

III. ADAPTIVE DIRECTIONAL CUBIC CONVOLUTION INTERPOLATION

A. Adaptive threshold selection

From the above DCC process, obviously, the quality and accuracy of edge detection and interpolation heavily depend on the selection of threshold T and k. In [16], experiments are carried out based on 24 Kodak color images [17], to determine the average best T from 1 to 1.15 and k from 1 to 6, according to the average peak signal-to-noise ratio (PSNR). As the results reveal, the effect on DCC by different k seems limited, and PSNR gets maximized by $k=5$. On the contrary, when T varies, PSNR values change in a wide range, which indicates that the selection of parameter T has a remarkable influence on DCC. When T is fixed to 1.15, it will lead to large deviation among different types of LR images. For

instance, the original QFN package defect image is shown in Fig.3(a). Large quantity of non-edge pixels are mistakenly detected as strong edge pixels using DCC, as shown in Fig.3(b). These pixels on weak edges or in texture regions then will be interpolated as on strong edges, which will lead to degradation of the interpolated image especially on weak edges and in texture regions. In other words, DCC using fixed threshold T may introduces directional artifacts in certain texture regions and it lacks preservation of weak edges and texture regions.

Thresholding is an effective and widely used technique in image segmentation and edge detection, to separate objects from the background. Otsu thresholding[18] is one of the most well-known thresholding techniques because of its simplicity and efficiency. Otsu thresholding obtains the threshold that maximizes variance between classes.

Supposing that the global threshold t is used to separate the image into the object class with the pixel range of $[0, t]$ and the

background class with the pixel range of $[t+1, L-1]$, where L denotes the gray level. Let the number of pixel value i be n_i, and the total pixels number be $N = \sum_{i=0}^{L-1} n_i$. The probability p_i of the occurrence of pixel value i can be defined as in Eq.(11),

$$p_i = \frac{n_i}{N} \qquad (11)$$

The bi-level Otsu function is defined as the sum of variance of the object class and the background class, as in Eq.(12),

$$\sigma_B^2(t) = \omega_1(t)(\mu_1(t) - \mu_T)^2 + \omega_2(t)(\mu_2(t) - \mu_T)^2 \qquad (12)$$

where μ_T denotes the mean intensity of the input image, determined as in Eq.(13), μ_1 and μ_2 denote the mean levels of the two classes, determined as in Eq.(14) and Eq.(15), ω_1 and ω_2 denote the probability of the two classes, defined as in Eq.(16) and Eq.(17).

$$\mu_T = \sum_{i=0}^{L-1} i p_i \qquad (13)$$

$$\mu_1(t) = \sum_{i=0}^{t} i p_i / \omega_1(t) \qquad (14)$$

$$\mu_2(t) = \sum_{i=t+1}^{L-1} i p_i / \omega_2(t) \qquad (15)$$

$$\omega_1(t) = \sum_{i=0}^{t} p_i \qquad (16)$$

$$\omega_2(t) = \sum_{i=t+1}^{L-1} p_i \qquad (17)$$

The optimal threshold t^* is configured by maximizing the objective function using

$$t^* = arg \max_{0 \le t \le L-1} \{ \sigma_B^2(t) \} \qquad (18)$$

In order to increase the accuracy of strong edge pixels detection and edge direction determination, Otsu thresholding technique is employed and the adaptive threshold t^* by Eq.(18), is used to replace the fixed parameter T to compare the two orthogonal gradients in local neighborhood around the missing pixel. In such a way, Eq.(3) in the original DCC is replaced by Eq.(19).

$$\begin{cases} \text{if } G_{135°} - G_{45°} > t^*, \text{ the pixel } I_h(2i, 2j) \text{ is on } 45° \text{ strong edge} \\ \text{else if } G_{45°} - G_{135°} > t^*, \text{ the pixel } I_h(2i, 2j) \text{ is on } 135° \text{ strong edge} \\ \text{else the pixel } I_h(2i, 2j) \text{ is on weak edge or in texture region} \\ \text{end} \end{cases} \quad (19)$$

Edge detection results of the same QFN package defect image using adaptive threshold are shown in Fig.3(c). Compared with the original DCC, our proposed adaptive threshold selection method is able to detect strong edge pixels more efficiently.

(a)　　　　　　　　　(b)　　　　　　　　　(c)
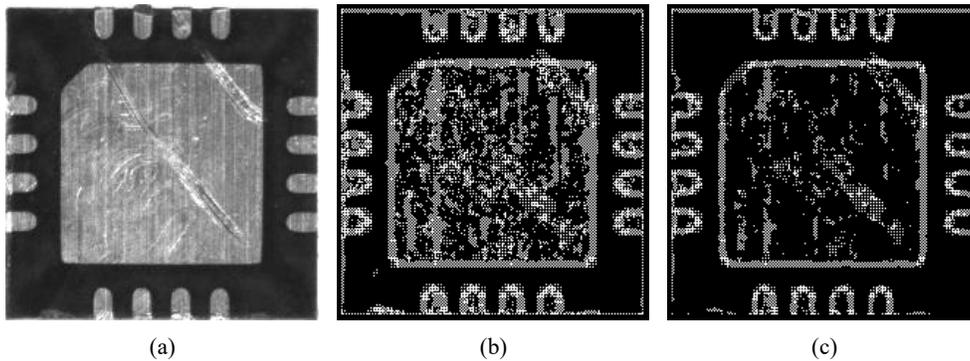
Fig.3. Comparison of edge detection using DCC and the proposed method; a) the original QFN package defect image; b) edges detected using DCC; and c) edges detected using the proposed method.

B. Method description

The boundary pixels haven't been taken into consideration in the original DCC. To address this issue, in the early stage of the proposed ADCC, boundary pixels are interpolated by the original bicubic algorithm [5], which will help improve the interpolation accuracy of the interior pixels. The detailed description of the proposed ADCC method for a scaling factor of 2 are given as follows. The interpolation for a scaling factor 2^n can be achieved by performing the following method by n times, iteratively.

Step 1. Calculate the global optimal threshold t^* for the original LR image I_1, as in Eq.(18);

Step 2. Expand the original LR image I_1 with the size of

$H \times W$ into the HR image I_h with the size of $(2H\text{-}1) \times (2W\text{-}1)$, where $I_1(i, j) = I_h(2i-1, 2j-1)$, $1 \le i \le H$, $1 \le j \le W$;

Step 3. Interpolate the boundary pixels using the original bicubic algorithm [5];

Step 4. Perform interpolation stage 1. Perform edge detection and then pixel value calculation of all the missing pixels $I_h(2i, 2j)$, as in Eq.(1), Eq.(2), Eq.(19), Eq.(4), Eq.(5), Eq.(6);

Step 5. Perform interpolation stage 2. Perform edge detection and then pixel value calculation of all the missing pixels $I_h(2i, 2j+1)$ and $I_h(2i+1, 2j)$, similar to Step 4, as in Eq.(7), Eq.(8), Eq.(19), Eq.(9), Eq.(10), Eq.(6);

Step 6. Output the HR image I_h with all the missing pixels interpolated.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed ADCC will be compared with the other four representative interpolation method in this section, which are bilinear [4], bicubic [5], NEDI [14] and the original DCC [16], both subjectively and objectively. All the five methods are implemented and experiments are conducted in MATLAB R2015b on a personal computer with 2.9 GHz CPU, 16 GB RAM. Six test images are used to conduct the experiments, including Kodak03, Kodak08, Kodak14 from Kodak Gray Image Set [17], and three typical QFN package defect images, denoted as QFN1, QFN2, QFN3, as shown in Fig.4.
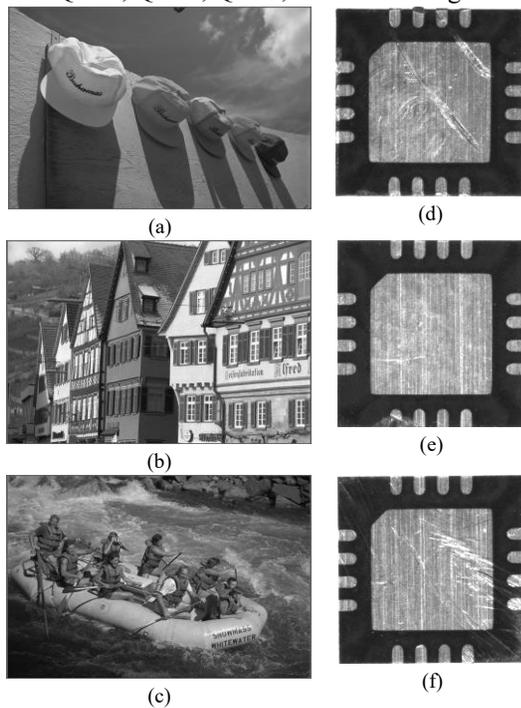
Fig.4. Test images; a) Kodak03 (768×512); b) Kodak08 (768×512); c) Kodak14 (768×512); d) QFN1 (214×216, the same as Fig.3(a)); e) QFN2 (212×212); and f) QFN3 (214×214).

A. Qualitative Analysis

To evaluate and compare the visual effects of the interpolated images using different interpolation methods, Kodak08 and QFN1 are downsampled with a factor of 4 and then interpolated to the original size. Local images of the two interpolation results using different methods are shown in Fig.5 and Fig.6, marked with red boxes.

It can be obtained from the local images of the interpolation results that bilinear and bicubic can preserve weak edges and texture regions, such as the house windows regions in Kodak08, black plastic encapsulated package and central bonding pad regions in QFN1. However, blurry and blocking artifacts are introduced around the diagonal edges, such as the house roof regions in Kodak08 and the diagonal scratch defect regions in QFN1, as shown in Fig.5(c), (d) and Fig.6(c), (d).

The NEDI preserves edges better than bilinear and bicubic but suffers a certain level of pixilation effect, as shown in Fig.5(e), and Fig.6(e). The DCC and the proposed ADCC are better able to preserve edges and structural information, and have better visual quality than the other three methods. However, several pixels on weak edges and in texture region, such as the house roof regions in Kodak08 and central bonding pad texture regions in QFN1, are detected and interpolated as strong edges and thus slight directional artifacts and ringing effect can be observed in DCC, as shown in Fig.5(f), and Fig.6(f). Compared with the other four methods, the proposed ADCC provides the best natural-looking image interpolation results, in terms of detail preservation and edge smoothness.
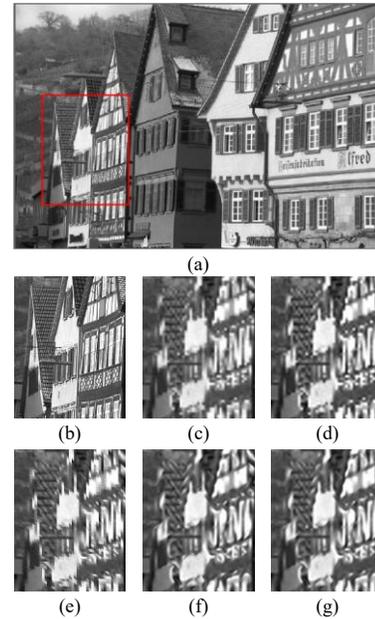
Fig.5. Interpolation results of Kodak08; a) the original image; b) its original local image; c) bilinear; d) bicubic; e) NEDI; f) DCC; and g) ADCC.
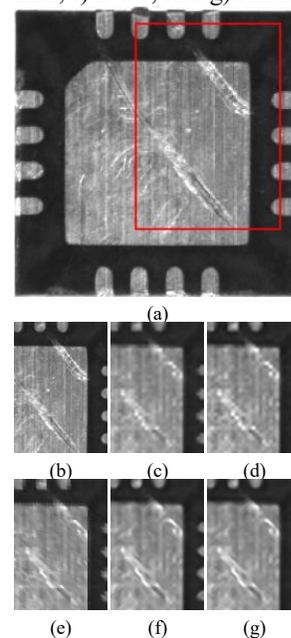
Fig.6. Interpolation results of QFN1; a) the original image; b) its original local image; c) bilinear; d) bicubic; e) NEDI; f) DCC; and g) ADCC.

B. Quantitative analysis

Since the original HR images of the six test images are known, PSNR and structural similarity (SSIM)[19] can be measured to evaluate the performance of the five interpolation methods objectively, via the original HR images and the interpolated HR images, which are similar to the qualitative analysis but a downsampling factor of 2.

PSNR is defined as in Eq.(20),

$$PSNR(dB) = 20\lg(\frac{255}{RMSE}) \qquad (20)$$

where

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - \tilde{I}(i,j)]^2} \qquad (21)$$

and I, $\tilde{I}$ are the original HR image and the interpolated image, with the size of $M \times N$, respectively. A higher PSNR value indicates a better interpolation quality.

SSIM is defined as in Eq.(22),

$$SSIM(x,y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma \qquad (22)$$

where x and y are the original HR image and the interpolated image, α, β and γ are positive constants to adjust the relative importance of the three components $l(x,y)$, $c(x,y)$ and $s(x,y)$, which denote the luminance comparison, the contrast comparison and the structure comparison, defined as in Eq.(23), Eq.(24), Eq.(25), respectively,

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \qquad (23)$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \qquad (24)$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \qquad (25)$$

where μ_x, μ_y are the mean intensity, σ_x and σ_y are the standard deviation, σ_{xy} is the correlation coefficient of the two images. $C_1 = (K_1L)^2$, $C_2 = (K_2L)^2$ and $C_3 = (K_3L)^2$ are three small positive constants, where L denotes the gray level and $K_1 \ll 1$, $K_2 \ll 1$, and $K_3 \ll 1$ are three small constants. A higher SSIM value indicates better structural similarity.

To reduce the boundary effect in the original DCC, described in section 3, the border of four pixels are excluded during the PSNR and SSIM calculation. The experimental results of PSNR, SSIM and computational time are shown in Table 1, Table 2 and Table 3, respectively. The best results are presented in bold.

From Table 1 and Table 2, the proposed ADCC significantly outperforms bilinear, bicubic, NEDI in terms of PSNR and SSIM, and outperforms the original DCC with slight margin, which indicates that the proposed ADCC achieves better interpolation quality, less distortion and better structural similarity for both classic test images and IC chip defect images.

In terms of computational time, from Table 3, bilinear and bicubic present lower computational time among other methods, but at the expense of poor interpolation quality in terms of both edge and texture preservation. Considering the fact that the proposed ADCC performs the boundary pixels bicubic pre-interpolation and Otsu thresholding, the time gap between DCC and ADCC is slight and acceptable.

As the application of the proposed ADCC in practical vision inspection for IC chip packages, IC chip images are interpolated online for a scaling factor of 2, e.g. QFN chip images from the size of 212×212 to the size of 413×413, to improve the perceptual quality and provide more details of defect regions. In the follow-up inspection process, the interpolated HR IC chip images indeed improve the accuracy and efficiency of the IC chip defect detection and classification.

Table 1. PSNR (dB) results of the interpolated HR images

Image	Bilinear	Bicubic	NEDI	DCC	ADCC
Kodak03	33.600	33.690	31.675	34.186	**34.187**
Kodak08	22.480	22.356	19.624	22.465	**22.485**
Kodak14	28.376	28.479	25.490	28.733	**28.746**
QFN1	28.472	28.476	25.814	28.587	**28.612**
QFN2	28.203	28.149	24.806	28.231	**28.284**
QFN3	26.700	26.650	24.477	26.778	**26.810**

Table 2. SSIM results of the interpolated HR images

Image	Bilinear	Bicubic	NEDI	DCC	ADCC
Kodak03	0.9693	0.9692	0.9641	0.9707	**0.9711**
Kodak08	0.9004	0.8994	0.8438	0.9009	**0.9010**
Kodak14	0.9407	0.9435	0.9114	0.9440	**0.9442**
QFN1	0.8365	0.8376	0.7756	0.8392	**0.8404**
QFN2	0.8260	0.8246	0.7505	0.8242	**0.8250**
QFN3	0.8091	0.8080	0.7462	0.8089	**0.8107**

Table 3. Computational time (s) of the interpolated HR images

Image	Bilinear	Bicubic	NEDI	DCC	ADCC
Kodak03	**0.0126**	0.0171	8.8642	2.0708	1.9984

Kodak08	**0.0138**	0.0168	9.0952	1.8631	1.9538
Kodak14	**0.0162**	0.0229	9.6447	1.9861	1.9741
QFN1	**0.0053**	0.0065	1.0330	0.2209	0.2244
QFN2	**0.0043**	0.0047	0.9653	0.2219	0.2239
QFN3	**0.0069**	0.0068	1.0618	0.2237	0.2294

V. CONCLUSION

In this paper, an adaptive directional cubic convolution interpolation method for IC chip defect images is proposed. In the presented method, the fixed threshold parameter in edge detection in the original DCC is replaced by the adaptive threshold, which is calculated using Otsu thresholding technique, to distinguish strong edge pixels from weak ones and texture regions, and estimate the direction of strong edges, adaptively. Boundary pixels are pre-interpolated using the original bicubic interpolation method to help improve the interpolation accuracy of the interior pixels. The experimental results of both classic test images and IC chip defect images demonstrate that the proposed ADCC outperforms the competing methods in terms of edge and texture preservation, interpolation quality and natural visual effect of the interpolated images. The proposed method can provide high quality IC chip images for defect detection and has been successfully applied on practical vision inspection for IC chips. Future works should be focused on reducing the computational complexity and further improving the operational efficiency of the proposed method.

VI. ACKNOWLEDGEMENTS

REFERENCES

[1] I Gershman, JB Bernstein. Structural health monitoring of solder joints in QFN package. Microelectronics Reliability, 2012, v 52, n 12: 3011-3016.
[2] Y Chao, M Dai, K Chen, P Chen, Z Zhang. A novel gravitational search algorithm for multilevel image segmentation and its application on semiconductor packages vision inspection. Optik - International Journal for Light and Electron Optics, 2016, v 127, n 14: 5770-5782.
[3] A Singh, J Singh. Review and Comparative analysis of various Image Interpolation Techniques. In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), IEEE, 2019.
[4] JS Lim. Two-dimensional signal and image processing. Englewood Cliffs, NJ, Prentice Hall, 1990, 710 p., 1990.
[5] RG Keys. CUBIC CONVOLUTION INTERPOLATION FOR DIGITAL IMAGE PROCESSING. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1981, v ASSP-29, n 6: 1153-1160.
[6] H Hou, H Andrews. Cubic splines for image interpolation and digital filtering. IEEE Transactions on acoustics, speech, and signal processing, 1978, v 26, n 6: 508-517.
[7] A Amanatiadis, I Andreadis. A survey on evaluation methods for image interpolation. Measurement Science and Technology, 2009, v 20, n 10: 104015.
[8] L Zhang, X Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE transactions on Image Processing, 2006, v 15, n 8: 2226-2238.
[9] H Fei, S Jinfei, C Ruwen, Z Songqing, Z Zhisheng. Noise-insensitive and edge-preserving resolution upconversion scheme for digital image based on the spatial general autoregressive model. IET Image Processing, 2016, v 10, n 4: 280-288.
[10] M Davoudi, A Heidarian. Design of Real-time Image Processing System for Measurement and Calibration of PSP. WSEAS TRANSACTIONS ON SIGNAL PROCESSING, 2020, v 16: 1-11.
[11] D Khaledyan, A Amirany, K Jafari, MH Moaiyeri, AZ Khuzani, N Mashhadi. Low-Cost Implementation of Bilinear and Bicubic Image Interpolation for Real-Time Image Super-Resolution. In 2020 IEEE Global Humanitarian Technology Conference (GHTC), IEEE, 2020.
[12] R Mamdouh, H El-Bakry, Ae-d Riad, N El-Khamisy. Converting 2D-Medical Image Files "DICOM" into 3D- Models, Based on Image Processing, and Analysing Their Results with Python Programming. WSEAS TRANSACTIONS ON COMPUTERS, 2020, v 19: 10-20.
[13] Z Huang, L Cao. Bicubic interpolation and extrapolation iteration method for high resolution digital holographic reconstruction. Optics and Lasers in Engineering, 2020, v 130: 106090.
[14] X Li, MT Orchard. New edge-directed interpolation. IEEE transactions on image processing, 2001, v 10, n 10: 1521-1527.
[15] S Stankevich, M Popov, S Shklyar, K Sukhanov, A Andreiev, A Lysenko, X Kun, C Shixiang, S Yupa, Z Xing, S Boya. Subpixel-shifted Satellite Images Superresolution: Software Implementation. WSEAS TRANSACTIONS ON COMPUTERS, 2020, v 19: 31-37.
[16] D Zhou, X Shen, W Dong. Image zooming using directional cubic convolution interpolation. IET Image Processing, 2012, v 6, n 6: 627-634.
[17] CIPR. Test images. Volume 2018. http://www.cipr.rpi.edu/resource/stills/kodak.html: CIPR, 2002.
[18] N Otsu. Threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics, 1979, v SMC-9, n 1: 62-66.
[19] Z Wang, AC Bovik, HR Sheikh, EP Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 2004, v 13, n 4: 600-612.