An Enhanced Model for Inpainting on Digital Images Using Dynamic Masking

M. S. Rana, M. M. Hassan, and T. Bhuiyan
Daffodil International University, Dhaka, Bangladesh
Email: s.rana_cse@yahoo.com; maruf.swe@diu.edu.bd; touhidbhuiyan@gmail.com

Abstract —In the digital world, inpainting is the algorithm used to replace or reconstruct lost, corrupted, or deteriorated parts of image data. Of the various proposed inpainting methods, convolutional methods are the simplest and most efficient. In this paper, an enhanced inpainting model based on convolution theorem is proposed for digital images that preserves the edge and effectively estimates the lost or damaged parts of an image. In the proposed algorithm, a mask image is created dynamically to detect the image area to inpaint where most of the algorithms detect the missing parts of the image manually. Studies confirm the simplicity and effectiveness of our method, which also produces results that are comparable to those produced using other methods.

Index Terms—Restoration, inpainting, filtering, convolution and PSNR

I. Introduction

Inpainting, the technique of changing an image in an undetectable form, is as antique as art itself. The areas and applications of inpainting are plentiful, from the restoration of damaged paintings and photographs to the removal/replacement of selected object. Reconstruction of missing or damaged portions of images is an ancient practice used broadly in artwork restoration. Also, known as inpainting or retouching, this activity consists of filling in the missing areas or modifying the damaged ones in a non-detectable way by an observer not familiar with the original images. Applications of image inpainting range from restoration of photographs, films and paintings, to removal of occlusions, such as text, subtitles, stamps and publicity from images. Inpainting is an artistic synonym for image interpolation, and has been circulated among museum restoration artists for a long time.

As an ancient painting gets older, the pigments in certain parts start falling off the canvas, and the painting becomes incomplete. The human act of filling in the missing parts of a painting is called "inpainting", as first introduced to image processing by Bertalmio, Sapiro, Caselles, and Ballester at the University of Minnesota (SIGGRAPH 2000) [1]-[3]. Digital inpainting has much wider applications in image processing and computer vision.

Inspired by the work of Bertalmio et al. (1999), we intend to develop general inpainting models for non-texture images.

Manuscript received January 23, 2017; revised April 24, 2017.
doi:10.12720/jcm.12.4.248-253

II. Related Works

Bertalmio *et al*. [1]-[3] pioneered a digital image inpainting algorithm based on partial differential equations (PDEs). A 2-D Laplacian is used to locally estimate the variation in color smoothness that is propagated along the isophote direction. The algorithm runs a few diffusion iterations to smooth the inpainted region after every few steps of the inpainting process. Anisotropic diffusion [4]-[6] is used to preserve boundaries across the inpainted region. This method requires difficult implementation processes and non-trivial iterative numerical methods such as anisotropic diffusion and/or multi-resolution schemes. The major limitations of this technique are as follows:

- Due to the small size of the isophotes, a greater number of iterations are needed for a larger inpainting area.
- As the isophote lines may cross each another, the inpainting loop has to be interleaved by anisotropic diffusion and more iterations, leading to blurring and resulting in inpainting error.
- The algorithm is inherently numerically unstable as it involves normalization of the gradient and Laplacian.

III. Inpainting Algorithm

A. Fundamental

Let Ω stand for the region to be inpainted, and $\delta\Omega$ for its boundary. The technique will prolong the isophote lines arriving at $\delta\Omega$ while maintaining the angle of "arrival". It proceeds from $\delta\Omega$ into this way, while curving the prolongation lines progressively to prevent them from crossing each other. Conservators at the Minneapolis Institute of Arts were consulted for this work, who clarified that inpainting is a subjective procedure and the underlying methodology is as follows [1]:

- Determining the global picture as to how to fill in the gap, the purpose of inpainting being to restore the unity of the work;
- The structure of the area surrounding Ω is continued into the gap, contour lines are drawn via the prolongation of those arriving at $\delta\Omega$.
- The different regions inside Ω, as defined by the contour lines, are filled with colors matching those inside $\delta\Omega$;

- The small details are painted; in other words, “texture” is added.

B. *Bertalmio’s Algorithm*

Let us assume that image I is of size M × N. Let (i, j) be the pixel location inside the inpainting region Ω, and Ω is the area to be inpainted [1]. Now estimate 2D smoothness estimation (L) using the Laplacian equation.

$$L^n(i,j) = I^n_{xx}(i,j) + I^n_{yy}(i,j) \tag{1}$$

where $I_{xx}(i,j)$ and $I_{yy}(i,j)$ are second-order derivatives of image at pixel (i, j) : 1 ⩽ i ⩽ M,1⩽ j ⩽ N along axes x and y, respectively. The change in L along the direction is given by.

$$\beta^n(i,j) := \delta\vec{L}^n(i,j)\cdot\frac{\vec{N}(i,j,n)}{|\vec{N}(i,j,n)|} \tag{2}$$

where

$$\delta\vec{L}^n(i,j) := (L^n(i+1,j) - L^n(i-1,j), L^n(i,j+1 - L^n(i,j-1)) \tag{3}$$

The isophote direction is given by

$$\frac{\vec{N}(i,j,n)}{|\vec{N}(i,j,n)|} := \frac{(-I^n_y(i,j), I^n_x(i,j)}{\sqrt{(I^n_x(i,j))^2 + (I^n_y(i,j))^2}} \tag{4}$$

Let In (i, j) denote each of the image pixels inside the region Ω at the inpainting “time” n. Then, the inpainting equation with an improvement factor is given by

$$I^{(n+1)}(i,j) = I^{(n)}(i,j) + \Delta t\beta^n\,|\Delta I^n(i,j)|, \forall(i,j)\in\Omega \tag{5}$$

where is ΔI^n called slope-limited version of the norm of the gradient which added in order to provide stability ensures that extremely large values will not be obtained for I^n (i, j) and keeps the resulting image from ‘blowing up’. The slope-limited norm is given as following equation

$$|\nabla I^n(i,j)| = \begin{cases} \sqrt{(I^n_{xbm})^2 + (I^n_{xfM})^2 + (I^n_{ybm})^2 + (I^n_{yfM})^2}, \text{when } \beta^n > 0 \\ \sqrt{(I^n_{xbM})^2 + (I^n_{xfm})^2 + (I^n_{ybM})^2 + (I^n_{yfm})^2}, \text{when } \beta^n < 0 \end{cases} \tag{6}$$

Indices and specify the difference between the intensities of the current pixel and the one in the opposite direction or forward, on OX and OY coordinate axes. Indices and prompt the fact that the minimum or the maximum value between the attained result and 0 will be chosen. This method encloses a number of inpainting steps with anisotropic diffusion [4]-[6] steps.

C. *The Oliveira Method*

Based on the Bertalmio method, Oliviera *et al.* [7]-[10] have proposed an inpainting algorithm that involves deleting color information inside the mask, followed by edge detection for the occluded/damaged area. Starting from the pixels on the edge, a convolution operation is then applied, using a neighborhood centered on each contour pixel and one of the proposed kernels. This simplest version of the algorithm, also called Fast Digital Image Inpainting, can introduce artifacts (noticeable blurring) when Ω crosses the boundaries of high-contrast edges. Oliviera method takes the image to inpaint on selected region by convolving with averaging filter has a zero weight at the center as shown in Fig. 1 with providing a simple pseudo code for execution process.

$$\begin{gathered} I'(i,j) = \sum_{i=0}^{M}\sum_{j=0}^{N} I(i,j).W1(i,j) \\ I_{out}(i,j) = \sum_{i=0}^{M}\sum_{j=0}^{N} I'(i,j).W2(i,j) \end{gathered} \tag{7}$$

where W1 and W1 are two diffusion kernels used with the algorithm. a= 0.073235, b = 0.176765, c = 0.125:

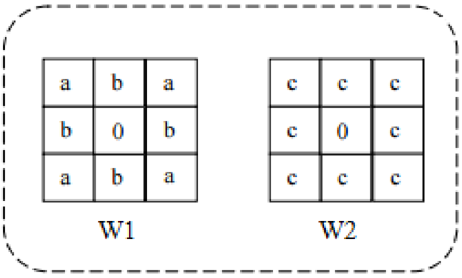

Fig. 1. Convoluting masks used by oliveira method

Pseudo code:

1. Initialize Ω
2. For (iteration=0; iteration <number_of_iteration; iteration ++)
3. If (check damaged region)
4. Convolve masked regions with kernel
5. End

D. *Hadhoud, Moustafa, and Shenoda’s Algorithm*

Hadhoud *et al*. [8]-[11] have proposed an improvement on the Oliveira method with respect to both the final image and the required processing time. Some steps have been retained from the original method, involving the selection of the mask and the removal of the existing color information in the mask. Unlike the Oliveira algorithm, the method uses a differently defined convolution kernel by using more known neighbors, and the restoration process can be achieved even within a single iteration.

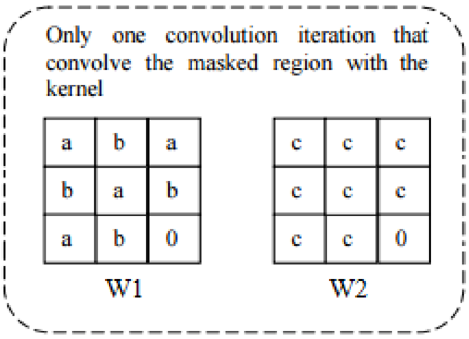

Fig. 2. Convoluting masks used by hadhoud method

In certain aspects, the modified method is identical to the Oliveira method, as it also involves selecting the region to be inpainted manually and the region to be inpainted is initialized by clearing its color. But the methods are different with respect to their respective filling methods, but different in the method of filling, Hadhoud method takes the image after the region to be inpainted selected and convolving the region with averaging filter has a zero weight at the bottom right corner instead of the center as show in Fig. 2.

E. The Convolution-Based Method

The main idea of the convolution-based inpainting algorithm, propounded by H. Noori, S. Saryazdi, and H. Nezamabadi-pour [12]-[15], is to use an adaptive kernel permitting a better processing edge regions. To do this, it uses the gradient of known pixels in the neighborhood of a missed pixel to compute weights in convolving mask. According to zero or small weights to missed pixels' neighbors with large local gradients, edges will be preserved better. Thus, the algorithm can estimate missed pixels while preserve sharp edges in image. The tricky is to design a function that is reciprocal to the gradients for smooth estimating and edge preserving. It proposes a function F(x) as follows to compute weights from the image gradient:

$$F(x) = \begin{cases} 1-(\frac{x}{\alpha})^2 & if\ |x| \le \frac{\alpha}{2} \\ (\frac{x}{\alpha}-1)^2 & if\ \frac{\alpha}{2} \le |x| \le \alpha \\ 0 & if\ |x| \ge \alpha \end{cases} \tag{8}$$

where x is gradient value of the current pixel in the image, α is a parameter giving an estimation of the missed pixel gradient and it control the quietness of propagation. The algorithm is as follows. Firstly, selecting a missed pixel on boundary of the damaged region, next considering a neighborhood around it and central gradients for each recognized pixel in the mask, is then calculated, finally, compute the weights as follows:

$$w(x) = \frac{1}{n} F(x_k) \tag{9}$$

where k presents the pixel position and w is the kernel weight at k, and n is the number of known pixel in the current neighborhood.

Finally, a value for a damaged pixel is calculated as follow

$$f^{/}(p) = (1-\sum_{k-1}^{n} w(k))\, f(p) + \sum_{k-1}^{n} w(k))\, f(k) \tag{10}$$

where f'(p) is the estimated value, f(k) is the value of a known pixel in the current neighborhood, n is the number of known pixels in the current neighborhood and k presents the pixel position and w is the kernel weight at k.

IV. Proposed Method

Regarding the procedure developed by Oliveira and its edition proposed by Hadhoud preserving edges is one of the major difficulties. In the case of Oliveira, defined some diffusion fences over the contour in order to stop the isotropic diffusion process; otherwise, some noticeable blurring effects may happen and in case of Hadhoud method, redefining the kernel and the direction of propagation leads to even more highlighted blurring effects and the loss of contour lines.

As an alternate to the 2-pixel width barriers defined according to Oliveira's idea, we are proposing an edge conserving technique by redefining the convoluting mask that encompasses the contour. In the proposed inpainting method, a damaged image is inpainted without blurring the output in a shorter span of time. Our proposed method dynamically detects the inpainted region by removing noise from the image. We use only one kernel/window W, whereas the Oliveira method uses two kernels/windows. We replace the values of the kernels/windows as follows:

0.080000	0.170000	0.080000
0.170000	0.080000	0.170000
0.080000	0.170000	0

Fig. 3. Kernel defined by the proposed method

A. Simulation

Based on Oliviera and Hadhoud, we have proposed an inpainting method [16]-[21] that involves deleting damaged information inside the mask (see Fig. 3), followed by edge detection (cf. the Canny edge detector) and smoothing the image to reduce the number of connected components for the damaged area Ω, as well as reducing image noise using a median filter. We start from the pixels on the edge. A convolution operation is applied using a neighborhood centered on each contour pixel and one of the proposed kernels. The algorithm can inpaint an image in just a few seconds.

Finally, we use the convolution equation as follows.

$$I_{out}(i,j) = \sum_{i=0}^{M}\sum_{j=0}^{N} I(i,j).W(i,j) \tag{11}$$

where $\boldsymbol{I_{out}}$ is the inpainted image, $\boldsymbol{I}$ is the input image, W is the mask kernel, and M x N is the size of $\boldsymbol{I}$.

B. Algorithm & Pseudo Code

The algorithm is as follows:

1. Input image with damage information
2. Color image converted into grayscale
3. Filter the image to remove noise using a median filter

$$\begin{aligned} W_n[1:F] &= I[1:MN] \\ I_{sort} &= sort\ (W_n) \\ I_{mid} &= mid\ (I_{sort}) \end{aligned} \tag{12}$$

4. Find edges of $\boldsymbol{I_{mid}}$ using the Canny edge detector and smooth the image to reduce the number of connected components producing $\boldsymbol{I_{canny}}$
5. Resize the image to produce mask image $\boldsymbol{I_{mask}}$ that gives the target region to be inpainted from $\boldsymbol{I_{canny}}$
6. Calculate connected components to extract all the connected components
7. Create 3x3 kernel $\boldsymbol{W}$ and convolve the image using the mask image and the following equation by checking damage using $\boldsymbol{I_{mask}}$

$$I_{out}(i,j)=\sum_{i=0}^{M}\sum_{j=0}^{N}I(i,j).W(i,j) \qquad (13)$$

8. Print the output and compute the execution time and PSNR.

C. Comparison Study with Table

Although the resulting family of images can be described as a combination among the original images and inpainted images. From the comparison table, we can see that the proposed modified method gives the better result of PSNR reducing the processing time as well as produce the result image without blur.

Fig. 4a. Input and output image damaged with cross lines

Fig. 4b. Input and output image damaged with text

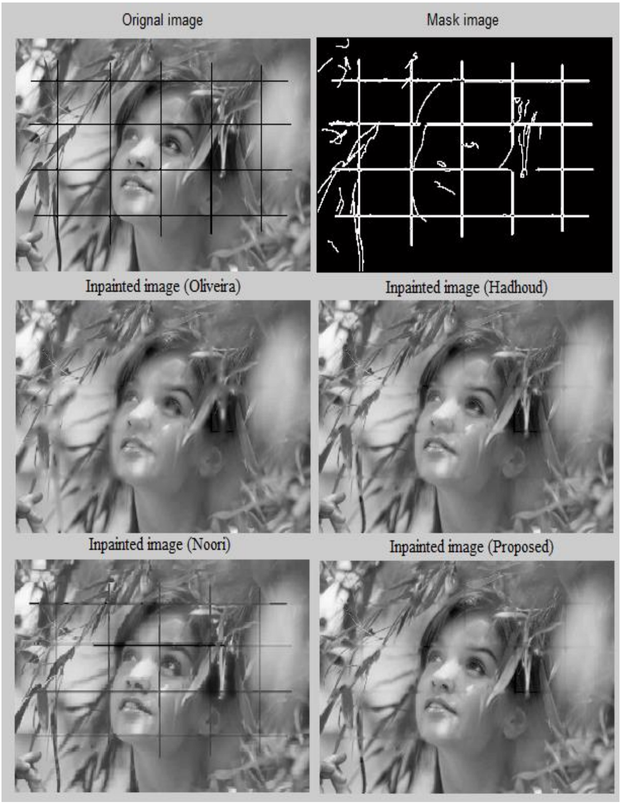

Fig. 4c. Input and output image damaged with rectangle lines

V. Experiment and Results

Various images were used in the experiment. The performance of the system was set and implemented in MATLAB. The experiment took place in several phases. Initially, the RGB image was converted into grayscale. The inpainting method restored the damaged image using different inpainting techniques. In brief, the experiment was conducted to inpaint a variety of damaged images and to record the results achieved by it.

A. Experimental Method and Results

The elementary prerequisite for all image inpainting processes [8], [16]-[21] is that the region to be inpainted should be selected manually by user, because no mathematical equation is capable of detecting or knowing the region to be inpainted without taking desired area. Because detecting the damaged or lost portion of image's area is very important. Our system aims at the automatic detection target area to be inpainted by creating mask. That means our method automatically generates mask image without user interaction that contains only target regions to be inpainted. As a result, the user intervention is simplified and the results are satisfactory without blurring output. In order to evaluate the performance of the proposed algorithm, in this section several comparative experiments are conducted. We applied the proposed algorithm, as well as those proposed by Bertalmio, Oliveira, and Hadhoud, to several damaged images of different contents. The experiments are done on an Intel(R) Core(TM) i5-4200M CPU @ 2.50 GHz 64-bit PC. The qualitative results are shown in Fig. 4a, Fig. 4b, and Fig. 4c, with comparison tables Table I and Table II. In Figure 4a, where the image is destroyed with several cross lines, the Oliveira, Hadhoud, and Noori algorithms lead to blurring, but the proposed algorithm achieves a satisfactory result. In Fig. 4b and Fig. 4c, the proposed algorithm provides the best results of all presented algorithms. Finally, the proposed method has the following advantages:

- Reducing the time of the inpainting process because of decreasing the number of iterations more than 100 times;
- Producing results without blurring and reducing noise;
- Dynamically detect the region to be inpainted.

TABLE I: COMPARISON STUDY OF EXECUTION TIME OF PROPOSED METHOD WITH OTHERS

Figure/ Method	Oliveira (seconds)	Noori (seconds)	Hadhoud (seconds)	Proposed (seconds)
Fig. 4a	98.2739	86.0736	9.6012	27.0833
Fig. 4b	42.7693	37.4856	7.1990	15.2771
Fig. 4c	77.6711	63.4485	9.0671	21.4831

TABLE II: COMPARISON STUDY OF PSNR OF PROPOSED METHOD WITH OTHERS

Figure/ Method	Oliveira	Noori	Hadhoud	Proposed
Fig. 4a	12.0129	7.0215	12.9902	13.2219
Fig. 4b	12.7494	9.1248	14.4092	15.1507
Fig. 4c	15.1130	6.2251	13.1350	13.0310

VI. CONCLUSIONS AND FUTURE WORK

During the last couple of years, a certain number of inpainting methods have been proposed, but it is still difficult to determine the appropriate one and also important to determine the algorithm parameters that lead to the best PSNR results and selecting representative images to provide relevant information. We have proposed a simple convolution based model which faster than others' algorithm with creating a dynamic kernel to detect the damaged area to be inpainted. Our proposed method can also substitute or restore the background when removing the large object from the image by removing noise without blurring the image. Our proposed modified method provides good results for 2D images but not for 3D images. The images used in the experiment had a "jpeg" extension. We would like to use our proposed method on 3D and 4D images, as well as image types like CT, MRI, and X-Ray.

REFERENCES

[1] M. Bertalmio, *et al.*, "Image inpainting," in *Proc. of SIGGRAPH 2000 Computer Graphics Processings*, 2000, pp. 417–424.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, *IEEE Trans. Img. Proc.*, vol. 10, pp. 1200-1211, 2001.

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proc. ICCV*, 2001, pp. 1335-1362.

[4] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, July 1990.

[5] M. M. Rahman, M. S. Rana, M. A. Islam, M. M. Rahman, and M. H. Talukder, "A new filtering technique for denoising speckle noise from medical images based on adaptive and anisotropic diffusion filter," *IJRCCT*, vol. 2, no. 9, September 2013.

[6] J. Weickert, *Anisotropic Diffusion in Image Processing*, Teuber Verlag, Stuttgart, 1998.

[7] M. Oliveira, B. Bowen, R. Mckenna, and Y. S. Chang, "Fast digital image inpainting," in *Proc. VIIP*, 2001, pp. 261–266.

[8] R. Vreja and R. Brad, "Image inpainting methods evaluation and improvement," *The Scientific World Journal*, vol. 2014, p. 11, 2014.

[9] F. Hollaus, "Different methods for image inpainting," Vienna University of Technology.

[10] J. (Jackie) Shen, "Inpainting and the fundamental problem of image processing," *SIAM News*, vol. 36, no. 5, June 2003.

[11] M. M. Hadhoud, K. A. Moustafa, and S. Z. Shendoa, "Digital images inpainting using modified convolution-based method," *International Journal of Signal Processing, Image Procesing and Pattern Recognition*, 2005.

[12] H. Noori, S. Saryazdi, and H. Nezamabadi-Pour, "A convolution based image inpainting," in *Proc. 1st International Conference on Communications Engineering*, 22–24, December 2010.

[13] H. Noori, S. Saryazdi, and H. Nezamabadi-Pour, "A bilateral image inpainting," *IJST, Transactions of Electrical Engineering*, vol. 35, no. E2, pp. 95-108, 2011.

[14] R. Kamran, M. Nasri, H. Nezamabadi-pour, and S. Saryazdi, "A new vector method for color image inpainting," in *Proc. First National Conference on New Ideas in Electrical Engineering*, 2012.

[15] R. L. Biradari and V. V. Kohir, "A novel image inpainting technique based on median diffusion," *Indian Academy of Sciences*, vol. 38, pp. 621–644, August 2013.

[16] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of Graphics Tools*, vol. 9, no. 1, 2004.

[17] T. F. Chan, J. Shen, and L. Vese, "Variational (PDE) models in image processing," *Notices Amer. Math. Soc.*, vol. 50, pp. 14–26, 2003.

[18] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200-1212, 2004.

[19] S. Tyagi and S. Singh, "Image inpainting by optimized exemplar region filling algorithm," *International Journal of Soft Computing and Engineering*, vol. 2, no. 6, Jan. 2013.

[20] S. Akhter and T. Tabassum, "Image completion with structure and texture inpainting," *International Journal for Advance Research in Engineering and Technology*, vol. 1, 7, Aug. 2013.

[21] V. V. Mahalingam, "Digital inpainting algorithms and evaluation," University of Kentucky UKnowledge, 2010.

Md. Shohel Rana was born in Sirajganj, Rajshahi, Bangladesh, in 1987. He received the B.Sc. (Engg.) and M.Sc. (Engg.) degree both from the Mawlana Bhashani Science and Technology University, Tangail, Bangladesh, both in Computer Science and Engineering in 2009 (Held 2010-11) and in 2014 respectively. Currently working as Senior Lecturer in the Department of Software Engineering, at Daffodil International University. Research interests include digital image processing and computer vision, human computer interaction, e-learning/remote learning, and web technology.

Md. Maruf Hassan was born in Bangladesh, in 1983. He received his Bachelor Degree in Information Systems and M.Sc. (Engg.) in Computer Science and Engineering. Currently, he is working as Assistant Professor in the Department of Software Engineering, at Daffodil International University. Research interests include digital image processing and computer vision, Information Security; Cyber Security.

Dr. Touhid Bhuiyan is the Founder Director of the Cyber Security Center, Daffodil International University (DIU), Bangladesh. His research interests are in cyber security, intelligent recommendations, social network, trust management, database management, e-Learning etc. He is the recipient of Australian Postgraduate Award and Deputy Vice-Chancellor's Initiative Scholarship from QUT, Australia. Before joining at DIU, he has employed by several renowned organizations including The University of Western Australia, Central College Sydney, Queensland University of Technology and University of Western Sydney. He has more than 18 years' experience in teaching, research and working at the IT industry in Australia, Singapore, Malaysia and Bangladesh.