Issue 4, Vol.3 (May 2014)
ISSN 2249-6149

RECONSTRUCTION OF TEXTURE FOR GRAY SCALE IMAGES USING ERROR REDUCTION ALGORITHM AND FFT ESTIMATION

Harish S.S.Velingkar[#1],Veena Devi S.V[#2]
#1 Borim Ponda-Goa,09591244431.
#2 Dept.Of ECE,RVCE,Bangalore,08762206268.

ABSTRACT

A method for reconstruction of missing texture using an iterative technique of Error Reduction(ER) algorithm & Fast Fourier Transform estimation has been explained in this brief. In the method we try to recover the missing texture patches in the image with the help of the errors converged in the ER algorithm using the known intensities. We initially identify two regions, one with known texture characteristics and the other with texture degradation ie. with unknown texture characteristics. We then break down both these regions into a number of smaller blocks and try to estimate the unknown texture characteristics by iterative comparison of the FFT magnitude with each of the known blocks. We further recover the phase of the original image using ER algorithm again and finally implement a circular filter to obtain the required results. Metrics such as SSIM,MSE & PSNR will be used to determine the amount of reconstruction obtained using the above method.

Key words: Error Reduction (ER) algorithm, image reconstruction, texture analysis, Fourier estimation, phase retrieval.

INTRODUCTION

Image restoration is the technique of undoing the effects caused on various types of noises on the quality of an image. The degradation caused may be due to various factors like mis-focus of the camera, due to scattered or reflected lights from the environment, device noise, quantization noise etc. The approach used in this project is one of the most novel techniques to recover back an image and the results obtained have been much more efficient than the conventional methods. Before briefing about the actual algorithm we need to a fair idea about the various concepts related to image restoration. An overview of such concepts has been given below. The main aim behind any restoration method is to undo or compensate for various factors that affect the quality of an image. Degradation can be caused due to many factors such as motion blur, noise, out of focus camera scattered and reflected light, quantization noise etc. The first image restoration processes started in the early 1950’s and were mainly use to reduce the storage capacity of an image. Later restoration was used in various applications like scientific explorations, legal investigations, film making, consumer photography etc. Among one of the oldest techniques used for image restoration is the process of image ‘Inpainting’. Inpainting [2] in general refers to reconstruction of the lost or deteriorated parts of an image or a video by making use of the known parts. For example in a museum which stores old valuable paintings the task of maintaining the quality of the image is of utter importance. Such a task is generally given to a skilled art conservator or an art restorer. In the digital world inpainting which is also referred to as image/video interpolation refers to the application of sophisticated algorithms to the images in order to restore the corrupted parts of an image. Inpainting is most effective for smaller parts and small defects. Such techniques are mainly used in photography and cinema to reverse the degradations like cracks or dust spots in photographs. Inpainting in general can be categorized into 2 parts. (i) Textural inpainting [3] and (ii) Structural inpainting [4]. Structural inpainting uses the geometrical information about an image to fill in the missing parts. Such algorithms focus on the consistency of the geometrical structures and need to necessarily reconstruct the actual texture of the image. Thereby to reconstruct the actual texture we use texture based restoration techniques. Such algorithms make use of repetitive texture characteristics of the known parts to fill the texture of the unknown parts. Nowadays combined texture and structural based restoration techniques have been used in various applications. . In the proposed method, texture based image reconstruction has been used as restoring the texture is the main priority of this approach.

ERROR REDUCTION ALGORITHM

The phase retrieval problem is of paramount importance in various areas of applied physics and engineering. The state of the art for solving this problem in two dimensions relies heavily on the pioneering work of Gerchberg, Saxton, and Fienup. The Gerchberg–Saxton algorithm, as well as its descendent in the form of the error reduction algorithm [1], was the first widely used numerical scheme to solve this type of problem. In image reconstruction, wave front sensing, astronomy, crystallography, and in other fields one often wishes to recover phase, although only intensity measurements are made. To solve the problem of phase recovery we make use of algorithms like ER algorithm, which is one of the steepest descent algorithms. These algorithms involve iterative Fourier transformation back and forth between the object and Fourier domains and application of the measured data or known constraints in each domain. The Gerchberg-Saxton algorithm was originally invented in connection with the problem of reconstructing phase from two intensity measurement (and for synthesizing phase codes given intensity constraints in each of two domains). The algorithm consists of the following four simple steps:

(1) Fourier transform an estimate of the object.
(2) Replace the modulus of the resulting computed Fourier transform with the measured Fourier modulus to form an estimate of the Fourier transform
(3) Inverse Fourier transform the estimate of the Fourier transform
(4) Replace the modulus of the resulting computed image with the measured object modulus to form a new estimate of the object.

The above steps can be summarized with the help of the following equations:

(1) $$G_k(u) = |G_k(u)| \exp[i\,\phi_k(u)] = \mathcal{F}[g_k(x)] \quad (1)$$

(2) $$G'_k(u) = |F(u)| \exp[i\,\phi_k(u)] \quad (2)$$

(3) $$g'_k(x) = \mathcal{F}^{-1}[G'_k(u)] \quad (3)$$

(4) $g_{k+1}(x) = | f(x) | \exp[i \phi' k(x)]$ (4)

Where

Gk(u) in eqn (1) is the Fourier transform of the unknown quantity (the distorted patch in our case)
G'k(u) in eqn (2) is the combined estimate which will have the magnitude of the known quantity and the phase of the unknown quantity
$g'_k(x)$ in eqn (3) is the inverse Fourier transform of the estimate
$g_{k+1}(x)$ in eqn (4) is the estimate for ever k^{th} iteration.

To see that the distortion is filled properly we set in the following criteria given by eqn (5):

$$g_{k+1}(x) = g'k(x) \quad x \neq \gamma \qquad (5)$$

Where γ is the set of points where the constraint set is violated. In our case it will be the threshold value set for either black or white distortion. The iterations continue until the computed Fourier transform satisfies the Fourier-domain constraints or the computed image satisfies the object-domain constraints, then one has found a solution, a Fourier transform pair that satisfies all the constraints in both domains. The convergence of the algorithm can be monitored by computing the squared error. In the Fourier domain the squared error is the sum of the squares of the amounts by which Gk (u), the computed Fourier transform, violates the Fourier-domain constraints. Since G'k (u) was formedfrom Gk(u) bymaking the minimum changes to satisfy the Fourier domain constraints, the squared error can be expressed in eqn (6) as:

$$E^2_{Fk} = \sum_m \sum_n | g_{kf}(u,v) - g'_k(u,v) |^2 \qquad (6)$$

Where, $0< m< M$, $0< n <N$

M being the number of known patches and N being the number of unknown patches in this case.
$g_{kf}(u,v)$ is the Inverse Fourier transform of the unknown quantity (after filling).
$g'_k(u,v)$ is the combined estimate which will have the magnitude of the known quantity and the phase of the unknown quantity. These errors are stored in the form of an array and the location where the minimum error exits is detected. The patch at that location is that taken in as the best estimated and substituted in for the unknown part. In practice, the error-reduction algorithm usually decreases the error rapidly for the first few iterations but much more slowly for later iterations.

METHODOLOGY

The entire process has been made user friendly, where the user can first select the co-ordinates of the patch that has to be reconstructed (distorted patch).Once the co-ordinates are filled in the user has to specify the colour of the distortion. Since we have implemented the method on gray scale images, black & white are the two types of distortion that we have considered. Consequently two different thresholds are set (0.63 for white and 0.05 for black) and are loaded in according to the type of distortion specified. Similarly while reconstructing the patch the user has to mention the x-co-ordinates so that only the required part of the patch is filled in and not the entire clipped strip. The methodology followed for the implementation of the proposed method consists of 4 major steps:

A)Selection of the target patch(with distortion) and dividing it into square blocks of mxm pixels.

The gray scale image with distortion (either black or white) is considered, and the coordinates of the image consisting of textual distortion are obtained. Using these coordinates the target patch of wxh pixels is clipped of from the image, filling the clipped part with a null matrix. Next the target patch is further divided into 'U' square blocks of mxm pixels (m preferably equal to h or a multiple of it for better results).The values of these blocks which are initially stored in the form of a column are reshaped in the form of a cell. Each row of such a cell will consist of a matrix of mxm pixels, with the entire cell having 'U' such matrices. Also the remaining patch with known intensities is divided into blocks of size mxm and stored in the form of a cell, with a total of 'K' such blocks.

B)Apply Error reduction algorithm(FFT estimation) to each of the blocks.

The cell consisting of matrices of size mxm is loaded (ie. both the known and target patch).Now we first try and get an estimated value of the Fourier transform for each block. It must me noted that the reason for choosing an imaginary transform is to get the phase of each of the target blocks and substitute the magnitude of the best matched known block in order to get the reconstructed image.

Let K={K1,K2….Kn) be the known patches of size mxm and U={U1,U2…Um} be the target patches of size mxm. First we take the 2-d FFT magnitude of the known patches. Let $|F_{k1}(u,v)|$ be the estimated Fourier transform magnitude of the first known patch. Now we get the phase angle of the first patch of the target image say θu1.Next we form a estimated function as in eqn (7):

$$F_{es1}= | F_{k1}(u,v) | \exp(j.\theta_{u1}) \qquad (7)$$

Next we take the Inverse Fourier transform of eqn (7) and store the mean of all the values in variable g_1 (u,v).Now we have an estimated Fourier transform magnitude with the magnitude of the first known patch and the phase of the first target patch. We then take the IFFT and store the first estimated value as fes1.

Next we start the actual substitution of this estimated value in the parts of the target patch where there is distortion. In case there is white distortion (ie. maximum intensity value is 1) we set the threshold for filling in the patches near to 1 but not exactly 1.This is because some of the patches at the edges of the distortion may not have the magnitude of 1 yet have to be reconstructed. Once all the pixels of the patch are filled we now have the actual filled patch with FFT of F_{t1}.

This method is then iterated till all the known patches have their own estimation of the target patch.

Thereby in all every patch will have 'K' such estimations, 'K' being the number of known patches. Next we find the patch with minimum error with the corresponding estimated patch, and this is then stored as the final substitution for the target patch as in eqn (8):

$$E_{min}=\sum_{n}\sum_{m}\{F_{esn}-F_{tn}\}^2 \quad (8)$$

$$0 < n < K \,,\, 0 < m < M$$

Where K,M are the number of known and unknown patches respectively. Once all the patches have been filled, they are stored in a column matrix and are prepared for the next step of reshaping.

C)The cell is converted back to matrix format and the target patch is reshaped, back to the clipped position.

Once the entire target patch has been filled with the texture from the known part of the image, the coordinates of the clipped patch(both y & x) are located again and the patches are put back into position using sliding window protocol. Few of the pixels may be lost due to sliding window protocol hence resizing of the image back to original may be required.

D)Filtering the output image using a circular filter in order to get improved results. Find the MSE,PSNR,SSIM with respect to the original image.

Once the image has been reconstructed, for improved results, the image has been passed through a circular averaging 2d filter. It consists of a function H which acts as a correlation kernel. For example h = fspecial ('disk', radius) returns a circular averaging filter (pillbox) within the square matrix of side (2*radius+1).

Once the final reconstructed image has been obtained, 3 metrics have been used to compare the amount of reconstruction caused namely MSE (Mean Squared Error),PSNR (Peak Signal to Noise Ratio) and SSIM (Structural Similarity Index). SSIM is designed to improve on traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human eye perception. The difference with respect to other techniques mentioned previously such as MSE or PSNR is that these approaches estimate perceived errors. On the other hand, SSIM considers image degradation as perceived change in structural information. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene.

RESULTS OF SIMULATION

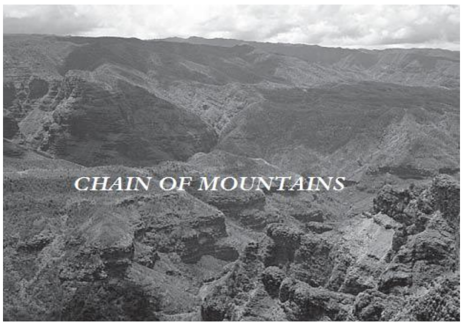

Fig 1. Distorted Gray scale image(480x360)

Fig 2. Reconstructed Image (Block size 18)

Fig 3. Reconstructed Image (Block size 10)

Table 1. Comparison of the above images (with different block sizes).Size of the distorted patch=18x380.It is seen that the best result is obtained when the block size equals width of the distorted patch.

Block size	18x18	10x10	29x29
Number of distorted patches	462	5057	371
Execution time of algorithm	1min	21min 34sec	47 sec
SSIM	0.97	0.978	0.96
MSE	1.37	1.83	5.22
PSNR	46.74	45.43	40.95
Remarks of the image quality	Good. Only little text border visible	Slight Blurring visible	Distortion border visible

The 3 images (of the same type) with different block sizes have been reconstructed and their results tabulated in Table 1. It can be seen clearly that when the block size is equal to the width of the distorted patch (ie.18) the result is most suitable with a SSIM of 0.97. When the block size is made less that the size of distorted patch (ie.10) the resultant reconstructed patch has to be stretched to fit the actual size & hence one can see the stretching effect. Also the MSE is higher than that for the block size of 18 and the PSNR is lower. When the block size is made greater than the size of distorted patch (ie.29) the distortion is slightly visible. Here there is a drastic increase in the MSE and considerable reduction in the PSNR which justifies the lowered quality. Thus it can be concluded that to get best result in terms of quality & execution time, the block size must be ideally kept equal to the size of the distorted patch.

Next, we consider and image which has black text distortion on 2 different lines. Both these lines will be considered in a single patch and the ER algorithm will be applied.

Fig 4. Distorted Gray Scale Image(640x480)

Fig 5. Reconstructed Image (Block size 33)

Fig 6. Reconstructed Image (Block size 20)

Table 2. Comparison of the above images (with different block sizes).Size of the distorted patch=33x480.It is seen that the best result is obtained when the block size equals width of the distorted patch.

Block size	33x33	20x20	53x53
Number of distorted patches	608	8694	588
Execution time of algorithm	1min 28sec	1hr 13min	2min 13 sec
SSIM(ideally=1)	0.986	0.989	0.97
MSE(ideally=0)	1.45	0.78	2.66
PSNR(ideally~50)	46.49	49.1	43.87
Remarks of the image quality	Very good. Distortion not visible	Mild Blurring visible	Distortion border visible

.The 3 images (of the same type) with different block sizes have been reconstructed and their results tabulated in Table 2. In this case although the distortion text is on 2 different lines once single patch has been considered which includes both lines & goes through the ER algorithm unlike the previous case where the two lines were clipped of separately. However the results are consistent with the above section. When the block size is equal to the width of the distorted patch (ie.33), best results are obtained (SSIM=0.98). When the block size is reduced (ie.20), slight stretching is visible hence affecting the image quality. In case of a block size bigger than the actual distortion (ie.53) the border of the distortion is visible. There is a considerable increase in the MSE and a decrease in PSNR which justifies the degraded image quality. This case however is most suitable when both the text lines are considerably close to each other.

Next we describe a GUI model which helps the user to load the distorted image, specify the type of distortion and compare the reconstructed image using the specified metrics.

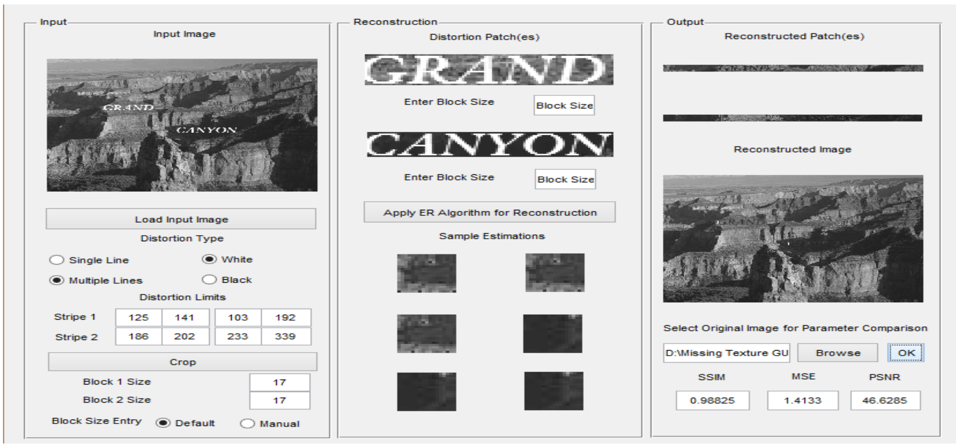

Fig 7. Final GUI Implemented

Fig. 7 represents the actual GUI (Graphical User Interface) that will be used by the user while restoring the image. The GUI is grouped into 3 parts: The first part is where the image is read, the distorted patch is cropped by the user, type of distortion is selected (either black or white) and the block size entered. If the user does not opt to manually enter the block size, the system will take the patch width as the default block size. The next section is where the clipped patch is displayed and ER algorithm is executed. Once the ER algorithm is executed, four sample patches of the filled distorted patch will be displayed as shown. In the last section the reconstructed patch as well as the restored image will be displayed. Next the user can browse the original image from the system (if available) and check the amount of reconstruction that has occurred. The 3 metrics namely SSIM, MSE, PSNR will be displayed in the 3 edit boxes as shown below.

CONCLUSION

An efficient algorithm for image restoration has been implemented and its results tabulated. The following points can be concluded from the same.

•For image reconstruction, use of a complex transform like FFT is much more efficient than comparison between the intensities

•The algorithm works effectively for gray scale images of any size, and fairly uniform texture characteristics (without too many edges/corners).

•For best results the block size must be equal to the width of the distorted patch. In case the block size is less, stretching is required to fit in the clipped patch back in the original image

•When the block size is halved, the execution time increases by about 20 times, which is not practical for real time applications. Hence again its best to keep the block size equal to the width of the distorted patch.

•The threshold for white & black distortion must also be selected carefully to get best restored outputs. In our case we have set white to 0.63 and black to 0.05.

REFERENCES

[1]. Ogawa,T.Haseyama, "Missing Texture Reconstruction Method Based on Error Reduction Algorithm Using Fourier Transform Magnitude Estimation Scheme", IEEE Trans. Image Processing, Vol. 22,issue 3,pp:1252-1257,July 2013.

[2]. Takahiro Ogawa, Miki Haseyama, and Hideo Kitajima, "Reconstruction method of missing texture using ER algorithm", IEEE International Conference of Image Proccessing (ICIP),pp:1026-1029,June 2005.

[3]. D. A. Karras, G. B. Mertzio. "Discretization Schemes and Numerical Approximations of PDE Inpainting Models and a comparative evaluation on novel real world MRI reconstruction applications," IEEE lnternational Workshop on Imaging Systems and Techniques(IST),pp:153-158, 14 May 2004.

[4]. T. Amano and Y. Sato, "Image interpolation using BPLP method on the eigenspace," Journal of System Computing(JSC), Vol. 38, no. 1, pp. 87–96, Jan. 2007.

[5]. T. Ogawa and M. Haseyama, "POCS-based texture reconstruction method using clustering scheme by kernel PCA," Institute of Electronics, Information and Communication Engineers (IEICE) Transaction Fundam., Vol. E90-A, no. 8, pp. 1519–1527, Aug. 2007.

[6]. J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for colour image restoration," IEEE Trans. Image Process., Vol. 17, no. 1, pp. 53–69, Jan. 2008.

[7]. I. Drori, D. Cohen-Or, and H. Teshurun, "Fragment-based image completion," in Proc. Special Interest Group on Graphics and Interactive Techniques(SIGGRAPH), pp. 303–312,Jan 2003.

[8]. Antonio Criminisi, Patrick Pérez, and Kentaro Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting", IEEE Transactions on image processing, Vol. 13, No. 9,pp:1200-1212, September 2004.

[9]. Tsz-Ho Kwok, Hoi Sheung, and Charlie C. L. Wang, Member, IEEE, "Fast Query for Exemplar-Based Image Completion" , IEEE Transactions on image processing, Vol. 19, No. 12, pp:3106-3115,December 2010.

[10]. Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," IEEE Trans. Image Process., Vol. 19, no. 5, pp. 1153–1165, May 2010.