



Restoration of Texture for Gray Scale Images by Error Reduction Algorithm using FFT Estimation

¹Harish S. S. Velingkar, ²Veena Devi S.V

^{1,2}Electronics & Communication Department, RVCE, Bangalore

Email:¹harishvelingkar@gmail.com,²veenadevi@rvce.edu.in

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 our 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.

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

I.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 [3] 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 [4] and (ii)Structuralinpainting[2]. 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.

II FUNDAMENTALS OF ER 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 [12], 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. One is usually interested in determining an object, $f(x)$, which is related to its Fourier transform, $F(u)$, by

$$F(u)=|F(u)|\exp(i\psi(u)) \quad = \quad \mathcal{F}[f(x)]$$

$$= \int_{-\infty}^{+\infty} f(x) \exp(-j2\pi ux) dx \quad (1)$$

where x is an M -dimensional spatial coordinate, and u is an M -dimensional spatial frequency coordinate. For the majority of interesting problems $M = 2$. In practice one deals with sampled data in the computer, where for the 2-D case, assuming square arrays, $u = (u_1, u_2)$ and $x = (x_1, x_2)$, where u_1, u_2, x_1 and $x_2 = 0, 1, 2, \dots, N-1$. Then one uses the discrete Fourier transform (DFT)

$$F(u) = \sum_{x=0}^{N-1} f(x) \cdot \exp\left(-\frac{j2\pi ux}{N}\right) \quad (2)$$

And its inverse,

$$F(x) = N^{-2} \sum_{u=0}^{N-1} F(u) \cdot \exp\left(\frac{j2\pi ux}{N}\right) \quad (3)$$

Which are computed using the FFT method described in the later sections.

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:

$$1] G_k(u) = |G_k(u)| \exp[i \phi_k(u)]$$

$$= \mathcal{F}[g_k(x)] \quad (4)$$

$$2] G'_k(u) = |F(u)| \exp[i \phi_k(u)] \quad (5)$$

$$3] g'_k(x) = \mathcal{F}^{-1}[G'_k(u)] \quad (6)$$

$$4] g_{k+1}(x) = |f(x)| \exp[i \phi'_k(x)] \quad (7)$$

Where

$G_k(u)$ is the Fourier transform of the unknown quantity (the distorted patch in our case)

$G'_k(u)$ is the combined estimate which will have the magnitude of the known quantity and the phase of the unknown quantity

$g'_k(x)$ is the inverse Fourier transform of the estimate

$g_{k+1}(x)$ is the estimate for ever k th iteration.

To see that the distortion is filled properly we set in the following criteria:

$$g_{k+1}(x) = \begin{cases} g'_k(x) & x \neq \gamma \\ 0 & x = \gamma \end{cases} \quad (8)$$

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 $G_k(u)$, the computed Fourier transform, violates the Fourier-domain constraints. Since $G'_k(u)$ was formed from $G_k(u)$ by making the minimum changes to satisfy the Fourier domain constraints, the squared error can be expressed as:

$$E_{Fk}^2 = \sum_m \sum_n |G_k(u) - G'_k(u)|^2 \quad (9)$$

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

M being the number of known patches and N being the number of unknown patches in our case.

These errors are stored in the form of an array and the location where the minimum error exists 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.

III USE OF FFT AND ITS PURPOSE

An image by itself is a real quantity and thus to transform the same into spectral domain we have to make use of a complex transform. The need for the use of a complex transform is because of two reasons. Firstly the need to recover the phase of the unknown quantity which in our case is the distorted part of the image. Secondly it has been verified that a FFT helps to recognize and differentiate between two different types of texture with less ambiguity than by comparing its intensities.



Fig 1.1 A gray scale image with multiple texture characteristics

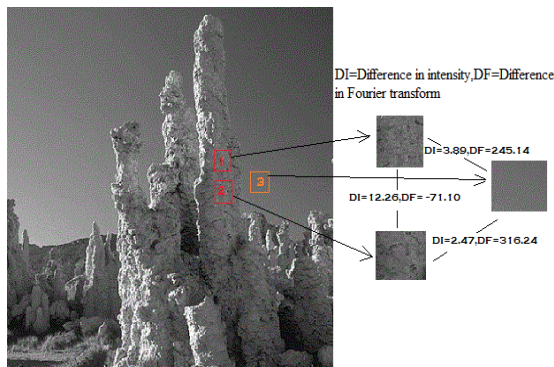


Fig 1.2 Example confirming the use of Fourier transform for texture comparison

Figure 4.2 shows a gray scale image which has different texture characteristics within itself.

We have then extracted 3 patches, 2 of which (patch 1 & 2) have the similar kind of texture and the third (patch 3) one with a different kind of texture. We then have taken the individual Fourier transform of each of these patches and subtracted them from one another. It can be seen that when we consider only DI i.e. the difference in intensities, there is not much a difference between patches with similar texture as well as patches with two different texture characteristics. Thus in an application like texture reconstruction where it is very much necessary to clearly distinguish between two kinds of texture without any ambiguity, just finding the difference of intensities is not sufficient. However when we take the difference of the Fourier transforms it can be seen that the patches can be differentiated easily without any ambiguity. Hence in our proposed method, FFT (fast Fourier transform) has been considered

In order to find the FFT the 'fft2' function is used in MATLAB. The 'fft2' functions are based on a library called FFTW. To compute a N-point DFT where $N=(N_1N_2)$, the library uses the Cooley-Tukey algorithm by default. Other algorithms like prime factor algorithm, split radix algorithm etc can also be used. If 'N' is a prime number, FFTW library decomposes an N-point problem into $3(N-1)$ point problems using Rader's

algorithm and then uses Cooley-Tukey for $(N-1)$ point DFT.

IV.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 $m \times m$ 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 $w \times h$ 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 $m \times m$ 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 $m \times m$ pixels, with the entire cell having 'U' such matrices. Also the remaining patch with known intensities is divided into blocks of size $m \times m$ 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 $m \times m$ is loaded (i.e. both the known and target patch). Now we first try and get an estimated value of the Fourier transform for each block. It must be 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 $K=\{K_1, K_2, \dots, K_n\}$ be the known patches of size $m \times m$ and $U=\{U_1, U_2, \dots, U_m\}$ be the target patches of size $m \times m$. 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 θ_{u1} . Next we form an estimated function

$$F_{es1} = |F_{k1}(u,v)| \exp(j\theta_{u1}) \quad (10)$$

Next we take the Inverse Fourier transform of eqn (1.10) and store the mean of all the values in variable

$g1(u,v)$. 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 fes_1 .

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.

$$E_{min} = \sum_n \sum_m \{Fes_n - Ft_n\}^2 \quad (11)$$

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

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 $h = \text{fspecial}('disk', \text{radius})$ returns a circular averaging filter (pillbox) within the square matrix of side $(2 * \text{radius} + 1)$.

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.

V RESULTS

A)



Fig 1.3 Distorted Gray scale image(640x480)



Fig 1.4 Reconstructed (Block size 16)



Fig 1.5 Reconstructed (Block size 8)

Table 1.1 Comparison of the above images (with different block sizes). Size of the distorted patch=16x480. It is seen that the best result is obtained when the block size equals width of the distorted patch.

Block size	16x16	8x8	32x32
Number of distorted patches	625	5697	609
Execution time of algorithm	2min 5sec	1hr 17min	1min 16 sec
SSIM(ideally=1)	0.98	0.99	0.97
MSE(ideally=0)	1.02	1.39	1.94
PSNR(ideally=50)	48.02	47.01	45.25
Remarks of the image quality	Very good. Distortion not visible	Blurring visible	Distortion border visible

B]

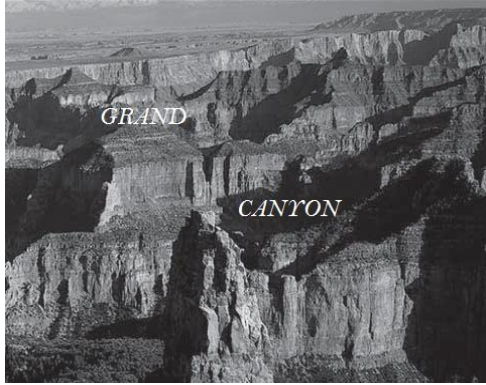


Fig 1.6 Distorted Gray Scale Image(480x360)



Fig 1.7 Reconstructed Image(Block size 17)



Fig 1.8 Reconstructed Image(Block size 32)

Table 1.2 Comparison of the above images (with different block sizes).Size of the distorted patch=17x360.It is seen that the best result is obtained when the block size equals width of the distorted patch.

Block size	17x17	9x9	32x32
Number of distorted patches	(480x2)=960	(4248x2)=8496	(449x2)=898
Execution time of algorithm	4min 59sec	1hr 16min	3min 18 sec
SSIM(ideally=1)	0.9884	0.9906	0.9815
MSE(ideally=0)	0.89	1.09	1.36
PSNR(ideally=50)	48.60	47.72	46.79
Remarks of the image quality	Very good. Distortion not visible	Mild Blurring visible	Distortion border visible

Table 1.1 Compares the 3 restored images of Sheer Rocky Cliff (distorted with black text) with varying varying patch sizes. The distorted patch in this ex.is of the size 16x480.Thereby it is seen that when the block size is equal to the width of the distorted patch, the best result with an SSIM of (0.98) is obtained. As the patch

size is decreased, the execution time exceeds feasible limits and while restoring, stretching of the patch to fill the actual distorted size is required. This stretching effect is readily visible in the output. When the block size exceeds the size of distorted patch, some of the distortion is visible which may be due to inclusion of unwanted edges and contrast shades in the selected patch. The results are consistent with the ones obtained in Table 1.2 where distortion on double lines is considered. Here the size of distorted patch is 17x360 and hence the best result of SSIM=0.98 is obtained when the block size is equal to width of the distorted patch. Stretching effect is noticed when the block size is halved and distortion is visible when the block size is greater than the width of the distortion.

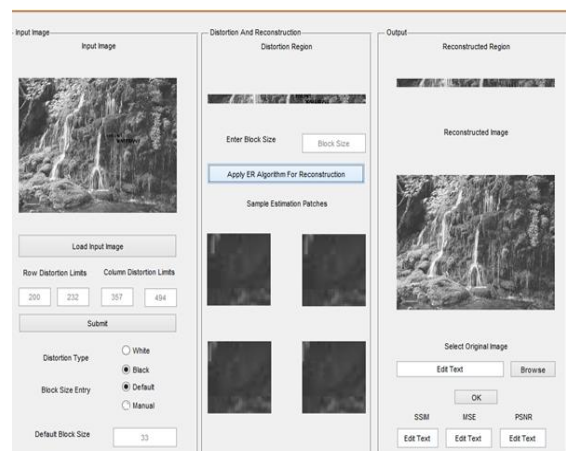


Fig 1.9 GUI Model to Test the Results

Fig 1.9 depicts the GUI model which can be used for single lined text, and was used for testing the effects of choosing various block sizes on the image. The coordinate values were chosen such that various block sizes are considered and the corresponding effect on the restored image was tabulated. Here the user has to manually enter the coordinate values (ie. X & Y Coordinates), select the type of distortion ie. Black or white and also the patch size required. In the end user can compare the restored image with the original image(if available) and obtain the required metric parameter values.

VI 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 Process, volume 22,issue 3,July 2013.
- [2]. C. Ballester, M. Bertalmio, V. Caselles, and G. Sapiro, "Filling-in by joint interpolation of vector fields and gray levels," IEEE Trans. Image Process., vol. 10, no. 8, pp. 1200–1211, Aug. 2001.
- [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 IST 2004 ~ International Workshop on Imaging Systems and Techniques,Strcsa, Italy, 14 May 2004.
- [4]. T. Amano and Y. Sato, "Image interpolation using BPLP method on the eigenspace," Syst. Comput. Jpn., 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," IEICE(Institute of Electronics, Information and Communication Engineers) Trans., vol. E90-A, no. 8, pp. 1519–1527, Aug. 2007.
- [6]. J.Mairal, M. Elad, and G. Sapiro, "Sparse representation for color 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. SIGGRAPH(Special Interest Group on Graphics) and Interactive Techniques,pp. 303–312. 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, September 2004.

