Image Compression Based on Spatial Redundancy Removal and Reconstruction using Image Inpainting

Veeramma Yatnalli 1, Prof. K L Sudha 2

1 JSS Academy of Technical Education
Bangalore, India
Email: veeramma71@gmail.com
2 Dayananda Sagar College of Engineering
Bangalore, India
Email: klsudha1@rediffmail.com

Abstract — In any digital communication system, the efficient transmission of an image over a channel involves source coding which represents the image in the compression form at the transmitter side. An information-carrying signal may be compressed by removing redundancy from the signal. In a lossless compression system statistical redundancy is removed so that the original signal can be perfectly reconstructed at the receiver. We present an image compression algorithm that integrates a novel technique for filling-in the missing information in an image. Firstly, the information in image regions is removed to reduce the data. The remaining data is transmitted and is used at the receiver’s side for recovering the removed regions. At the decoder, to retrieve removed regions, an inpainting method is applied using the information extracted from the received image.

Index Terms — Inpainting, Compression, Interpolation, Spatial Redundancy, Edge Detection

I. INTRODUCTION

Image compression has been a very active field of research and development and many different systems and algorithms for compression and decompression have been proposed and developed. In order to encourage interworking, competition and increased choice, it has been necessary to define standard methods of compression encoding and decoding to allow products from different manufacturers to communicate effectively. This has led to the development of a number of key International Standards for image and video compression, including the JPEG, MPEG and H.26x series of standards. Compression is the process of compacting data into a smaller number of bits. Compression involves a complementary pair of systems, a compressor (encoder) and a decompressor (decoder). The encoder converts the source data into a compressed form (occupying a reduced number of bits) prior to transmission or storage and the decoder converts the compressed form back into a representation of the original data. As far as the reversibility of the original image is concerned, the image compression algorithms can be broadly classified in two categories—lossless and lossy. The lossy compression techniques are usually applicable to data where high fidelity of reconstructed data is not required for perception by the human perceptual system. Examples of such types of data are image, video, graphics, speech, audio, etc. Image inpainting is an image interpolation technique that has many intense applications in image processing. However, to fill the lost information in an image is still a challenging problem. Image inpainting modifies and fills-in the missing areas of image in a non-noticeable way, by an observer not familiar with the original image [1]. The user provides a mask image representing the regions removed. The mask image contains black pixels representing the region
removed. The missing information in the image is then filled by propagating the pixels inwards from the boundary of the missing region. In section 2, we have discussed the related work done in this field. Section 3 describes our technique of image inpainting based on the interpolation problem. The performance measurement parameters are listed in section 4. Section 5 presents several results. Finally, the conclusions are presented in section 6.

II. RELATED WORK

The notion of image inpainting was first introduced by Bertalmio et al. [1]. The image smoothness information interpolated by the image Laplacian is propagated along the isophotes directions, which are estimated by the gradient of image rotated by 90 degrees. Later, Bertalmio et al. [2] presented a variational method of inpainting based on the joint interpolation of the gradient directions and the gray-levels. The system of PDEs (Partial Differential Equations) is formulated and is solved using level sets of the intensity function. Bertalmio et al. [3] introduced another method using the framework of the Navier-Stokes equations. The major problem with the above methods [1,2,3] is that they are unable to reconstruct textured regions. Masnou and Morel [4,5] proposed an inpainting algorithm based on level lines. This variational approach was based on the calculation of generalized elastica energy, and joining the level lines having the lowest energy of all compatible T-junctions.

For non-texture images, there exist other productive theoretical image models, which are well known as TV (Total Variation) model[7] and Mumford-Shah model[8]. Chan and Shen proposed two image inpainting algorithms [9, 10] based on the TV model. The TV inpainting model [9] is based on Euler-Lagrange equation and the model employs anisotropic diffusion [11] inside the inpainting domain. Outside the inpainting domain the equation denoises the image and makes the method sturdy against noise. Since this method failed to connect broken edges, Chan and Shen extended the TV algorithm to Curvature-Driven Diffusion (CDD) model [10]. In this method, the conductivity coefficient of the model depends on the curvature of the isophotes. This can connect some broken edges but the results are blurry. Another type of method given by Oliveira [13] inpaints by repeatedly convolving a filter mask with the inpainting domain. This solution behaves very similar to the solution of the linear diffusion equation. The algorithm tends to blur edges, due to inherent linear diffusion, unless user selects high-gradient image areas manually before inpainting. The above methods have several disadvantages. The implementation of PDE based methods require nontrivial iterative numerical techniques. Also, in some of them various threshold values are not given, and even some steps are mentioned as numerically unstable.

Another kind of work found in this field is related to texture synthesis. It basically deals with selecting texture and synthesizing it into the inpainting domain. Although some excellent results are demonstrated in the literature by some methods [14, 15, 16, 17], yet they require user to manually select the texture to be copied in the missing area. So, in the images which have some structures, would need a lot of work from user to find and mark the replacements for the hole. Other methods based on texture synthesis [18, 19, 20, 21, 22] try to reproduce the texture from a smaller source sample by copying color values from the source. They have difficulty in filling-in images with structures. The exemplar-based method given by Criminisi et al. [23] searches for a similar patch in whole image that consumes a lot of time while our algorithm searches in a limited window and gives comparable results.

III. IMAGE COMPRESSION BASED ON SPATIAL REDUNDANCY REMOVAL AND RECONSTRUCTION USING IMAGE INPAINTING

The algorithm for image compression using inpainting consists the following steps:
1. Apply Canny Edge Detector to identify the edges in an image.
2. Analyse the image carefully to remove the blocks.
3. Fill in removed blocks using Image Inpainting Technique
Steps 1 and 2 are carried out at the encoder side.
At the decoder side, step 3 is performed. The steps are summarized in Fig 1.

![Figure 1.Block diagram of proposed algorithm](image_url)

The three algorithms will be described in the following order.

A. Edge Detection
A process of identifying an edge is known as Edge detection. The sharp change in pixel intensity of image can be identified as the edge of the image. Edges of the image correspond to points, where the gray value changes significantly from one pixel to the next pixel. Edge detection significantly reduces the amount of data and removes redundant information, while preserving the important structural properties in an image. Edge detection using gradient method is very efficient. It is implemented in our work using Canny algorithm.

**Canny Edge Detection Technique**

The Canny edge detection algorithm is popularly known as optimal edge detector. Edge detection is important with these three criteria: The first criterion is that edges occurring in images should not be missed and no response to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. Based on these criteria, the Canny edge detector first smoothes the image to eliminate noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis.

**B. Block Removal:**

The image generated from the edge detection process is further analysed carefully for the purpose of block removal. Our goal is to remove as many blocks as possible and still recover the image with good perceptual quality. Remove the blocks in smooth area so that they can be restored back using structure inpainting. Such blocks are not noticeable even if we simply fill in DC value. We can apply structure inpainting if a block falls into one of the two cases: When a block does not contain 'strong edges', and when it is not composed of fine repetitive patterns. Sharp (strong) edges are critical when human recognize the shape of an object. Based on these points, only those blocks which do not have the edge pixels are removed and the blocks containing the critical edge points are retained since it is hard to recover them.

**C. Image Inpainting based on Interpolation**

The algorithm presented here allows interpolation from known values on the boundary of the missing domain in such a way that the interpolated values are “smooth”. The technique is based on solving the combinatorial Laplace equation with Dirichlet boundary conditions given by the known pixel values. The algorithm is independent of the size or the topology of the region to be filled. Digital Inpainting helps to perform inpainting digitally through image processing in some sense. Thereby, automating the process and reducing the interaction required by the user. The only interaction required by the user is the selection of the region of the image to be removed [10]. The user can select an area through a free hand selection or polygon selection. In practice, the area is selected using any image manipulation software and is given a different color. In our algorithm we have removed the different regions of the image from the knowledge of the edge detector. After the user selects the regions to be removed, the remaining portion of the image is transmitted. The inpainting algorithm automatically fills-in these regions with information surrounding them. The fill-in is done in such a way that isophote lines arriving at the regions’ boundaries are propagated inside. This is automatically done (and in a fast way), thereby allowing to simultaneously fill-in numerous regions containing completely different structures and surrounding backgrounds.

![Figure 2](image-url)

Figure 2 In the image $u_0$, the region $\Omega$ to be inpainted and its boundary $\partial \Omega$.

Now, let $\Omega$ denote the set of pixels (the region) of the image to be inpainted. Let $\partial \Omega$ denote the pixel boundary of $\Omega$ so that $\partial \Omega \subset \Omega$ as shown in Fig. 2.

Partial differential equations (PDEs) are used for a large variety of image processing tasks, and recently, they have been proposed for so-called inpainting techniques, which use PDE-based interpolation methods to fill in missing image data from a given inpainting mask. In inpainting, we find an information about the region to be redefined. This information is obtained from the regions surrounding the missing area. The following steps describe the general solution to the problem:

**STEP 1:** SPECIFY $\Omega$

**STEP 2:** $\partial \Omega = \text{THE BOUNDARY OF } \Omega$ (Region outside Missing Area)

**STEP 3:** INPAINT ALL PIXELS IN $\Omega$ BASED ON INFORMATION in $\partial \Omega$
Let us consider a 4x4 sub-block \( f(x, y) \) from the received image as shown below.

\[
\begin{pmatrix}
  z_1 & z_2 & z_3 & z_4 \\
  z_5 & z_6 & z_7 & z_8 \\
  z_9 & z_{10} & z_{11} & z_{12} \\
  z_{13} & z_{14} & z_{15} & z_{16}
\end{pmatrix}
\]

The 4x4 sub-block \( f(x, y) \) has the pixel values as shown below.

\[
f(x, y) = \begin{pmatrix}
  1 & 2 & 3 & 4 \\
  2 & 2 & 2 & 3 \\
  4 & 3 & 2 & 1
\end{pmatrix}
\]

In this image, a redundant information characterized by uniformly distributed gray level is removed. That is, the information about the pixels \( z_6, z_7, z_{10} \) and \( z_{11} \) has been totally lost and the resulting block is as shown below.

\[
f(x, y) = \begin{pmatrix}
  1 & 2 & 3 & 4 \\
  1 & 0 & 0 & 3 \\
  2 & 0 & 0 & 3 \\
  4 & 3 & 2 & 1
\end{pmatrix}
\]

If the symbol \( ? \) stands for the corresponding grey scale value of the pixel unknown, these pixels are redefined as NaN (Not a Number) elements or missing domain pixels. In other words, points for which the values are not fixed (i.e., missing data) are termed as interior points. In the sub-block \( f(x, y) \), the elements \( z_6, z_7, z_{10} \) and \( z_{11} \) are interior points or unknown points in the missing domain. The resulting sub-block \( f(x, y) \) is now referred as mask image having unknown and known boundary pixels. Then the matrix mask representing the above image is given by,

\[
f(x, y) = \begin{pmatrix}
  1 & 2 & 3 & 4 \\
  1 & ? & ? & 3 \\
  2 & ? & ? & 3 \\
  4 & 3 & 2 & 1
\end{pmatrix}
\]

\[
f(x, y) = \begin{pmatrix}
  1 & ? & ? & 3 \\
  2 & NaN & NaN & 3 \\
  4 & NaN & NaN & 3
\end{pmatrix}
\]

In an image, the pixels for which there exist a fixed value (i.e., outside the missing domain) are termed as boundary points. The set of boundary points provides a Dirichlet boundary condition. The problem of finding harmonic function subject to its boundary conditions is called the Dirichlet problem. The Dirichlet problem can be stated as follows:

*Given a function \( f \) that has values everywhere on the boundary of a region in \( \mathbb{R}^n \), is there a unique continuous function \( u \) twice continuously differentiable in the interior and continuous on the boundary, such that \( u \) is harmonic in the interior and \( u = f \) on the boundary. This requirement is called the Dirichlet boundary condition.*

Finding a harmonic function that satisfies the boundary conditions may be viewed as a method for finding values on the interior of the volume that interpolate between the boundary values in the smoothest possible fashion. Continuous function twice continuously differentiable in the interior can be written as \( \nabla^2 f(x, y) \) which can further be approximated in terms of finite difference. The function \( \nabla^2 f(x, y) \) is expressed as,

\[
\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]
where the function $f(x, y)$ describes a two-dimensional matrix. The discrete form of Laplacian operator in $x$ and $y$ direction is defined in below equation as,

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \quad (2)$$

In this way the interior points are connected to themselves with ‘4’ and they are connected to neighboring points with ‘-1’. This computation is performed for all the interior unknown points in the 4x4 sub-block $f(x, y)$, i.e., for all unknown points in a matrix $f(x, y)$ the equation 2 is applied. The resulting matrix $D$ is as shown in Fig. 3a, and sparse representation of the matrix $D$ is as shown in Fig. 3b.

To find the values of interior points, the solution of equations is obtained for $X$ as

$$D \cdot X = R$$

$D$ represents connection of an interior point to the neighbors and to itself. Where $X$ is (n x m) x 1 row matrix where, each row corresponds to one unknown pixel.

$R$ in the above equation is obtained from the pixel values at the boundary of the inpainting domain. Therefore, we can write the linear equation form using,

$$D \cdot X = R \quad \text{as} \quad X = R \setminus D$$

The set of linear equations are solved finally to obtain the values for NaN elements.

![Figure 3a Matrix D](image)

![Figure 3b Sparse Matrix of sum of row and column](image)

The set of linear equations are solved to finally obtain the values for NaN elements as shown below,

$$f'(x, y) = \begin{bmatrix}
1 & 2 & 3 & 4 \\
2 & 2.37 & 2.50 & 3 \\
4 & 3 & 2 & 1 
\end{bmatrix}$$

This examples is simply considered for the sake of illustration by considering a 4x4 sub-block $f(x, y)$ of an image $u$. The methodology can be applied for all the unknown elements and using the boundary information their values can be determined.

### IV PERFORMANCE EVALUATION

#### A. Measurement of Image Quality
The design of an imaging system should begin with an analysis of the physical characteristics of the originals and the means through which the images may be generated. A detailed examination of some of the originals may be necessary to determine the level of detail within the original that might be meaningful for a researcher or scholar. Generally image quality is assessed from Quality Assessment Parameters. The two commonly used measures for quantifying the error between images are Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

**B. Mean Square Error:** MSE between the original image $I$ and the reconstructed image $I'$ is given by

$$
MSE = \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} [I(x,y) - I'(x,y)]^2
$$

It is very useful measure as it gives an average value of energy lost in the lossy compression of the original image $I$. A human observing two images affected by the same type of degradation will generally judge the one with smaller MSE closer to the original. A very small MSE indicates that the image is very close to original.

**C. Peak Signal to Noise Ratio (PSNR)**

The PSNR between two $8 \times 8$ bit images (in dB) can be obtained using the formula, which tends to be cited more often, since it is a logarithmic measure, and our brains seem to respond logarithmically to intensity. Increasing PSNR represents increasing fidelity of compression. Generally, when the PSNR is 40 dB or larger, the two images are virtually distinguishable by human observers.

$$
PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right)
$$

A very logical way of measuring how well a compression algorithm compresses a given set of data is to look at the ratio of the number of bits required to represent the data before compression to the number of bits required to represent the data after compression. This ratio is called as **compression ratio**.

**V. RESULTS**

An increase in the compression performance can be obtained by removing the non-significant information in the image prior to compression and then performing image Inpainting at the receiver. The Inpainting technique is applied to images(256x256) with incomplete image received.

In the first step of our approach, to remove the blocks that can be restored at the decoder side, an edge detection is performed. The edge image tells us about the points which can be retained and are used for the transmission purpose. Based on this analysis, we further have removed the blocks in various regions so that they can be restored back using inpainting. We have used images of different characteristics for the algorithm. In Fig. 4, the mask images are generated without considering the edge information. This consideration allows to remove more information. Every alternate block in the image is removed. These incomplete/compressed images containing only 50% of the information are sent over the reliable communication channel. Using the information available in the received images, the missing information is restored back with the Inpainting algorithm. Various images like, Lena, Cameraman, Barbara and House are used in our work for the experimental purpose. The received images are shown in Fig. 4a, Fig. 4b, Fig. 4c, and Fig. 4d. The reconstructed images at the decoder side using the inpainting algorithms are as shown in Fig. 4e, Fig. 4f, Fig. 4g and Fig. 4f.

We can notice from the resulting images that, visual qualities of the image are not so good and the PSNR values are low. However, the compression ratio value are significantly good in comparison with the original size of the image as can be observed from the calculation. Table I illustrates the various quality assessment parameters implemented using Inpainting. In Table I, the quality assessment parameters like MSE, PSNR and Compression Ratio are tabulated. From Table I, we observe that the number of pixels in the compressed image is significantly less in comparison with the pixels in the original image. This increases the compression ratio at a greater extent.

To improve the visual quality and to preserve the edges properly, we have applied edge detection using Canny edge detector. The blocks that contain the edges are not involved in the process of block removal, i.e., any block containing the edge information is retained and the remaining blocks that do not contain the edges or the blocks with uniform pixel values are removed. The incomplete image is sent over any reliable communication channel. The received image is completed with the surrounding information available with the removed block using interpolation based Inpainting algorithm as explained in the previous section. As expected, the recovered image's visual quality is very good and the PSNR value is also improved significantly. The edge detected images are shown in Fig. 5a, Fig. 5b, Fig. 5c, and Fig. 5d.

The received images are shown in Fig. 5e, Fig. 5f, Fig. 5g, and Fig. 5h. The reconstructed images at the decoder side
using the inpainting algorithms are as shown in Fig. 5i, Fig. 5j, Fig. 5k and Fig. 5l. Table II illustrates the results that are obtained the image compression using Inpainting algorithm with spatial redundancy removed. We can notice from the resulting images that, visual qualities of the image has been improved significantly because of the fact that the edges were exclude in the process of block removal. The PSNR values are also increased relatively. However, the compression ratio value are reduced in comparison with the earlier approach. Table II illustrates the various quality assessment parameters implemented using Inpainting. In Table II, the quality assessment parameters like MSE, PSNR and Compression Ratio are tabulated.

<table>
<thead>
<tr>
<th>Image (256x256)</th>
<th>Compression achieved with Inpainting</th>
<th>Mean Square Error (MSE)</th>
<th>Peak Signal to Noise Ratio (PSNR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>1.5788</td>
<td>59.4573</td>
<td>30.3888</td>
</tr>
<tr>
<td>House</td>
<td>1.5788</td>
<td>25.6825</td>
<td>34.0344</td>
</tr>
<tr>
<td>Barbara</td>
<td>1.5788</td>
<td>39.1727</td>
<td>32.2010</td>
</tr>
<tr>
<td>Lena</td>
<td>1.5788</td>
<td>21.1551</td>
<td>34.8767</td>
</tr>
</tbody>
</table>
Figure 4 Various Results of Image Reconstruction using Image Inpainting
(a) Mask Image of Barbara (b) Mask Image of Cameraman
(c) Mask Image of House (d) Mask Image of Lena (e) Restored Image of Barbara (f) Restored Image of Cameraman
(g) Restored Image of House (h) Restored Image of Lena

<table>
<thead>
<tr>
<th>Image</th>
<th>Compression achieved with Inpainting</th>
<th>Mean Square Error (MSE)</th>
<th>Peak Signal to Noise Ratio (PSNR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cameraman</td>
<td>1.4147</td>
<td>2.8262</td>
<td>43.6188</td>
</tr>
<tr>
<td>House</td>
<td>1.4819</td>
<td>4.6414</td>
<td>41.4643</td>
</tr>
<tr>
<td>Barbara</td>
<td>1.3638</td>
<td>12.5616</td>
<td>37.1403</td>
</tr>
<tr>
<td>Lena</td>
<td>1.4152</td>
<td>5.7055</td>
<td>40.5679</td>
</tr>
</tbody>
</table>
Figure. 5 Various Results of Image Reconstruction using Image Inpainting
(a) Edge Image of Barbara (b) Edge Image of Cameraman (c) Edge Image of House (d) Edge Image of Lena (e) Mask Image of Barbara (f) Mask Image of Cameraman (g) Mask Image of House (h) Mask Image of Lena (i) Restored Image of Barbara (j) Restored Image of Cameraman (k) Restored Image of House (l) Restored Image of Lena

VI CONCLUSION

A technique for the filling-in of missing blocks in wireless transmission of compressed images is discussed. Here, we are dealing with filling—in of missing blocks which are intentionally removed before the transmission over the channel. This approach could be utilized to reconstruct the incomplete image even though a large amount of redundant information is removed prior to transmission. As long as the features in the image are not completely lost, they can be satisfactorily reconstructed using computationally efficient image inpainting algorithms. This eliminates the need for retransmission of lost blocks. When the image resolution is increased, the quality of reconstruction improves and a retransmission request is rarely required, resulting in a better effective data transmission rate. The future work in this field is to improve the compression, quality of the image and to utilize the bandwidth efficiently. If the textured blocks are also removed prior to transmission, we can achieve better compression performance with the texture synthesis algorithms

References: