**American Journal of Applied Mathematics**
2015; 3(1-2): 9-13
Published online December 26, 2014 (http://www.sciencepublishinggroup.com/j/ajam)
doi: 10.11648/j.ajam.s.2015030102.12
ISSN: 2330-0043 (Print); ISSN: 2330-006X (Online)

# Digital restoration of archaeological heritage

**Supriya Deshpande[1], Preeti Sangamesh[2]**

[1]Dept of CSE, PESIT, South Campus, Bangalore, India-560100
[2]Asst.Professor, Dept. of CSE, PESIT-South Campus, Bangalore, India – 560100

## Email address:
supriya.deshpande19@gmail.com (S. Deshpande), preetiyaragol@pes.edu (P. Sangamesh)

## To cite this article:
Supriya Deshpande, Preeti Sangamesh. Digital Restoration of Archaeological Heritage. *American Journal of Applied Mathematics.* Special Issue: Frontiers in Mathematics and Computing. Vol. 3, No. 1-2, 2015, pp. 9-13. doi: 10.11648/j.ajam.s.2015030102.12

**Abstract:** Virtual Restoration of archaeological heritage stems from the need to create a clearer and better image of the beautiful historic monuments now in ruins. It gives the viewer a sense and feel of how the heritage originally looked. For this, there are many restoration projects of the major historic sites, paintings etc, going on across the world. Computers have been introduced to archaeology and cultural heritage as tools for promoting scientific work and as electronic aids for providing users with substantial information on archaeological heritage. Small holes and breakages in the monuments/paintings can severely degrade its appeal to viewers. Image restoration is the operation of taking a corrupted/noisy image and estimating the clean original image. In this work, we have proposed a method to automatically detect the defect in the corrupted image using Perona Malik Anisotropic Diffusion and Binary Thresholding, followed by Image Inpainting with Navier-Stokes Method, which have been found to be effective in the art of restoring lost/selected parts of an image based on the background information in a visually plausible way.

**Keywords:** Automatic Detection of Defect, Image Inpainting, Image Restoration, Navier-Stokes Inpainting

## 1. Introduction

Today's society can very effectively be referred to as the 'Image Society'. This is not just because image is a powerful used medium of communication but also because it is an easy, compact and widespread way to represent the physical world around us. We rely on images. A huge amount of digital information is available today. The increase in capacity of computers and the advancement made in the image acquisition devices have led to this phenomenon [1].

Digital restoration of archaeological heritage stems from the need to create a more realistic image of the beautiful historic monuments, paintings etc, which are now in ruins. It gives the viewer a sense and feel of how the original was. It creates an allusion, a virtual replica of the original.

Traditionally, skilled craftsmen did the work of restoration manually. However, more recently, the use of computers to enable image restoration has gained a lot of interest. Various Image processing techniques have been employed over the time to get better and more accurate results when applying digital restoration.

The ancient practice of reconstruction of missing or damaged portions of images has been employed extensively in artwork restoration. This activity, also known as *inpainting* or *retouching*, consists of filling in the missing areas or modifying the damaged ones such that the missing area is unidentifiable by an observer not familiar with the original images.

Inpainting is the operation of taking a corrupted/noisy image and reconstructing the clean original image. Corruption may come in many forms such as motion blur, noise, damage, stains, breakages etc. Image restoration is applied to produce realistic data from a scientific point of view. Digital image restoration using inpainting has varied applications. It is used in restoring old photographs, archaeological heritage sites, monuments, paintings etc.

We propose a method wherein the defect in the image is automatically detected. This is accomplished using Anisotropic Diffusion and Binary Thresholding. The detected defect is taken as the mask for the Image Inpainting using Navier-Stokes Algorithm to restore lost/selected parts of an image of historical importance such that we get results very close to the original.

## 2. Related Work

Criminisi et al. present a method – called exemplar-based image inpainting [2]. It uses the main idea of PDE to fill in texture. This approach helps in filling bigger regions. The exemplar based technique fills the inpainting region with a texture, generated by texture patches from the surrounding areas. However, in this method the linear structures bordering the target region only influences the fill order. The method of exemplar-based image inpainting, thus combines the strengths of PDE based methods and exemplar-based techniques.

Alexandru Telea[3] proposed another method for inpainting using Fast Marching Algorithm (FMM). In this, the image smoothness is estimated as a weighted average over a known image neighborhood of the pixel to be inpainted. The missing regions are treated as level sets. The FMM algorithm is then used to propagate the image information across the level sets to fill up the missing information. A narrow distance between the known and the unknown pixels is maintained by the FMM. The FMM starts with inpainting the unknown pixel closest to the known pixel. Once the unknown pixel has been inpainted, its value is updated to the set of known pixels.

Bertalmio et al[4] coined the term Digital Image Inpainting, and was the first to present a digital image-inpainting algorithm based on partial differential equations (PDEs). In this approach, the user provides a mask provided specifying the portions of the input image to be retouched. The algorithm treats the input image as three separate channels (R, G and B). For each channel, it fills in the areas to be inpainted by propagating information from the outside of the masked region along level lines (*isophotes*). The algorithm also introduces the importance of propagating both the gradient direction (geometry) and gray-values (photometry) of the image in a band surrounding the hole to be filled-in. Isophote directions are obtained by computing at each pixel along the inpainting contour a discretized gradient vector.

# 3. Proposed Methodology

In many of the existing inpainting algorithms, a user-provided mask is required.

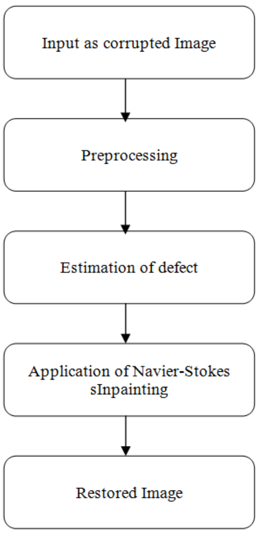


***Figure 1.*** *Block diagram of the proposed method*

In this work, we propose a method to get an automatic estimation of defects in the corrupted image, which is used to further do inpainting. Our work builds up on Bertalmio et al's Inpainting method using Navier-Stokes Equation. Our method estimates the defect area by using a series of steps, makes a binary mask and applies inpainting technique. Block diagram of the proposed system is shown in Figure 1.

*Preprocessing*

We provide the corrupted/damaged image as the input. Image preprocessing is the technique of enhancing data images prior to computational processing is performed. It is used to suppress unwanted distortions and noise, and to enhance important image features for further processing.

***Figure 2.*** *Defected Image of a site in Hampi, India*

In our method, we do preprocessing in two steps.

a. *RGB to Grayscale Conversion:* We convert the corrupted image, example as given in Figure2, into gray-scale as a part of preprocessing. This is done so that each image pixel carries only the information of the intensity where 0 corresponds to pure black, 255 to pure white, and all intermediate values to different shades of gray.

A digital grayscale image, I, is an m × n matrix, where at each index, $I_{i,j}$ has an integer value from 0 to 255. This value is referred to as the graylevel at location (i, j), where 0 corresponds to pure black, 255 to pure white, and all intermediate values to different shades of gray. The basic problem of grayscale transformation is to reproduce the intent of the color original, its contrasts and salient features, while preserving the perceived magnitude and direction of its gradients. The goal of converting a color image into grayscale is to create a perceptually accurate version of the color image that represents the exact image in grey tones to a viewer and removes all the unnecessary parts from the image if required.

b. *Anisotropic Diffusion:* In the next step we apply Perona-Malik's Anisotropic Diffusion[5] to the grayscale image. It is a technique which aims at reducing the image noise without removing significant parts of the image, image content, typically edges, lines or other details that are important for the interpretation of the image.

This technique resembles the process that creates a scale space, where an image generates a parameterized family of successively more and more blurred images based on a diffusion process. It produces a family of parameterized images, but each resulting image is a combination between the original image and a filter that depends on the local content of the original image.

Let $\Omega \in R^2$, denote a subset of plane I (.,t): $\Omega \in R$ , be a family of gray scale image

Then, anisotropic diffusion can be defined as:

$$\frac{\partial I}{\partial t} = div(C(x,y,t)\nabla I) = \nabla C.\nabla I + C(x,y,t)\Delta t \qquad (1)$$

Where,

$\nabla$ denotes the Laplacian,

$\Delta$ denotes the gradient,

div(…) denotes the divergence operator,

C(x,y,t) is the diffusion coefficient.

The equation (i) can be reduced to the heat equation if C(x,y,t) is constant.

$$I_t = C\Delta I \qquad (2)$$

By setting the conduction coefficient to 1 in the interior of the region and to 0 at the boundary of the region, we can ensure smoothing only within the region and not across the boundaries.

When conduction coefficient is chosen as a function of the magnitude of the gradient of the brightness function –

$$C(x,y,t) = g(\|\nabla I(x,y,t)\|) \qquad (3)$$

This preserves the edges of the image as well as sharpens the edges on correct choice of g(.)

In our approach, we use the function g(.) as

$$C(\|\nabla I\|) = e^{(-(\|\nabla I\|/K)^2)} \qquad (4)$$

where the constant K controls the sensitivity to edges. We choose k between 20 – 100, depending on the image requirements as on keeping kappa low, small intensity gradients are able to block conduction and hence we have diffusion across step edges. On keeping kappa to a large value the influence of intensity gradients on conduction is reduced.

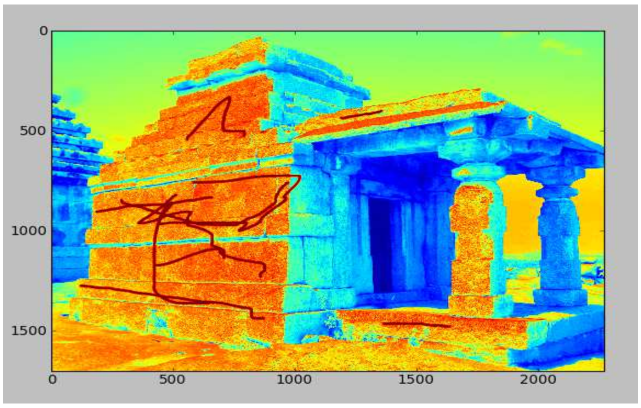


***Figure 3.*** *Anisotropic Diffusion on Corrupted Image of Figure 1*

We perform Anisotropic Diffusion iteratively around 3-5 times to sufficiently blur the interiors. We also use another parameter, gamma, to control the speed of diffusion, which is kept approximately between 0.1 – 0.2 to maintain the stability of the diffusion process. The result is as shown in Figure 3.

By using Anisotropic Diffusion, we get the sharp edges in the image. Experimentally, it has been noted down that the edge detection using the Perona-Malik Anisotropic Diffusion method outperforms other edge detection methods such as Canny Edge Detection[6][7].

Using this, we get an approximation of the defect area.

*Estimation of the defect area*

*Binary Thresholding*: After the application of Anisotropic Diffusion on our images, we get a result with blurred interiors and sharp edges. In the next step, we apply Binary Thresholding on the image. Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding is used to create binary images. The gray-level values below or equal to the selected threshold are usually classified as background, while the values above this threshold are classified as object.

Binary images are used as a mask of the defected part. It creates a mask represented in either white or black (binary).

Binary Thresholding on the anisotropically diffused image helps in automatically detecting the defected area. Binary thresholding creates a mask where the defect is marked as red, and the non defected area is marked in blue.

In our approach, we set the threshold value, which is used to classify the pixel intensities in the grayscale image to between 190 and 210. We also set a maximum threshold value to 255 for pixels in the image which cross the threshold value.

By using these as our parameters, we get a good estimate of the region where the defect lies as shown in Figure 4.

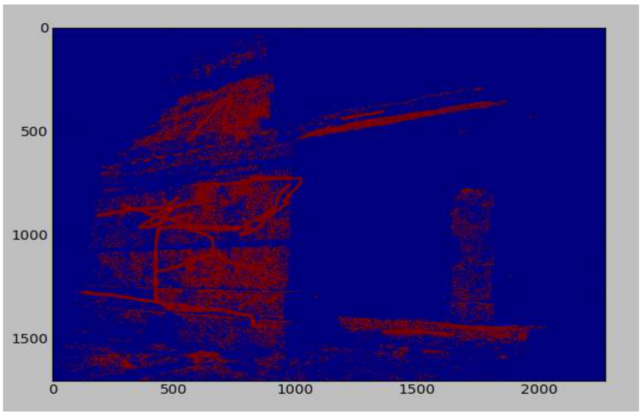


***Figure 4.*** *Damage Estimation on Binary Thresholding*

*Application of Navier-Stokes Inpainting*

The binary mask obtained from the previous steps is used as the mask. The corrupted part of the image is restored by inpainting on applying Navier-Stokes equation.

The Navier-Stokes Inpainting approach adapts the equation for fluid dynamics for images. The isophote lines are continuously propagated from the exterior of the region to be inpainted, to its interior.

Bertalmio et al, proposed to view the Image Intensity (**I**) as a 'stream function' (φ) for a two-dimensional incompressible flow. Vorticity (w=Δ**φ**) of the fluid is replaced by the Laplacian of the Image Intensity, called smoothness (w=Δ**I)**. The Image Intensity is transported into the region to be inpainted by a vector field defined by the stream function.

The algorithm continues along the isophote while matching gradient vectors at the boundary of the inpainting region. For a seamless filling, it is important to propagate both the gradient direction (geometry) and gray-values (photometry) of the image in a band surrounding the hole to be filled-in.

In order to obtain a smooth interpolation and continuation of isophotes, a high-order Partial Differential Equations (PDE's) or systems of PDE's needs to be performed. From the paper, Image Inpainting[8], we use an algorithm to project the gradient of the smoothness of the image intensity in the direction of the isophotes., which results in a discrete approximation of the PDE

$$I_t = \nabla^{\perp} I . \nabla \Delta I \tag{5}$$

Where,

$\nabla^{\perp}$ is the perpendicular gradient $(-\partial y, \partial x)$,

$\Delta$ is the Laplace Operator $\partial_x^2 + \partial_y^2$.

To get (5) to behave as a steady state solution, we perform,

$$\nabla^{\perp} I . \nabla \Delta I = 0 \tag{6}$$

Here, the image intensity propagates along the isophote lines.

When using any PDE-based method to do inpainting, the issue of boundary conditions becomes very important. To get results of inpainting which are imperceptible to the eye, continuation of the image intensity and direction of the isophote lines needs to be performed continuously into the inpainting region.

Therefore, PDE-based method involving the image intensity *I*, must enforce Dirichlet (fixed *I*) boundary conditions as well as a condition on the direction of $\Delta I$ on the boundary. However, lower order PDE's can enforce only one of these two boundary conditions for I. Thus, inpainting would have discontinuities in the slope of isophote lines. In Navier-Stoke's method, this is handled by using a vector evolution for ∇I.

The Navier-Stokes analogy guarantees, in a very natural way, continuity of the image intensity function I and its isophote directions across the boundary of the inpainting region. For the Navier-Stokes inpainting method, the continuity across the boundary also produces an image intensity function that is continuous across the boundary ( $\partial\Omega$) of the inpainted region.

On applying Navier-Stoke's inpainting algorithm to our example, we get the result as shown in Figure 5.

***Figure 5.*** *Inpainted Image*

Experimentation and Result

We implemented the method described in this paper in OpenCV with Python.

Another sample was taken from Shavanabelagola Temple, India. The results of which are as shown.

Figure 6 shows the damaged image of the statue in Shravanabelgola. Figure 7 shows the Anisotropically Diffused image. Figure 8 shows the estimation of the damaged area on Binary Thresholding. Figure 9 shows the final inpainted image.

***Figure 6.*** *Damaged Image of Statue*

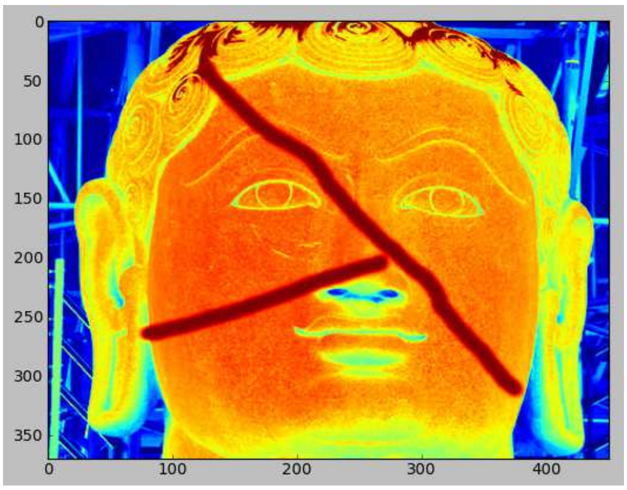


***Figure 7.*** *Anisotropic Diffusion on Image*

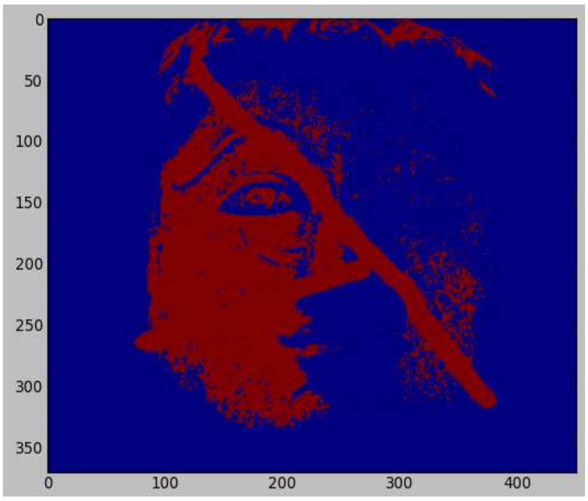


***Figure 8.*** *Defect Estimation on Binary Thresholding.*

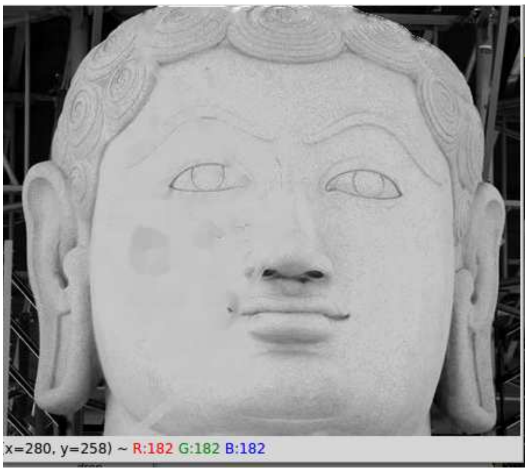


***Figure 9.*** *Inpainted Image*

The experimentation performed on 20 training samples showed a result of 99.3% correct estimation of the defect area using our method.

For grey-shade images, inpainting using Navier-Stokes gave a good final inpainted image. However, blurring around edges was observed.

The limitation of the inpainting algorithm applied using the estimated defect binary mask was that it does not work well with non-grey shade images, giving high amount of blurring across the edges as well as spreading of the multiple colours on the defect area.

## 4. Conclusion

In this paper we have proposed an efficient way to perform automatic detection of the defect area in damaged images. The damaged images are first preprocessed by being converted to greyscale. Further, Perona-Malik Anisotropic Diffusion is applied to the greyscale image, which removes all unwanted image noise from the image, while keeping all the significant image content like the edges/lines intact. Application of Binary Thresholding on the Anisotropically Diffused Image, finally gives us the correct estimate of the defect area of the image. This is further used as a mask for performing Navier-Stoke's Inpainting.

## References

[1] Gilles Aubert, Pierre Kornprobst, "Mathematical Problems in Image Processing Partial Differential Equations and the Calculus of Variations"

[2] P. P. A. Criminisi, K. Toyama, "Region filling and object removal by exemplar-based inpainting", in: IEEE Trans. Image Processing, vol. 13, no. 9, 2004, pp. 1200-1212.

[3] Atelea, Alexandru. "An image inpainting technique based on the fast marching method." Journal of graphics tools 9.1 (2004): 23-34.

[4] Bertalmio, Marcelo, Andrea L. Bertozzi, and Guillermo Sapiro. "Navier-stokes, fluid dynamics, and image and video inpainting." In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, pp. I-355. IEEE, 2001.

[5] Pietro Perona and Jitendra Malik (July 1990). *"Scale-space and edge detection using anisotropic diffusion"*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 (7): 629–639

[6] Joachim Weickert, "Anisotropic Diffusion in Image Processing".

[7] Canny, J., A "Computational Approach To Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence

[8] M. Bertalmio, G. Sapiro, C. Ballester and V. Caselles, "Image inpainting," Computer Graphics, SIGGRAPH 2000, pp. 417-424, July 2000.