Image Restoration Using In-Painting Techniques

Arnav Sankhe
Computer Engineering
Rajiv Gandhi Institute of Technology
Mumbai, India
arnav.sankhe19@gmail.com

Aman Sarwagi
Compute Engineering
Rajiv Gandhi Institute of Technology
Mumbai, India
aman.sarwagi@gmail.com

***Abstract*—Image in-painting refers to the process of restoring missing or damaged areas in an image. Image In-painting is used to filling the misplaced or smashed region in an image make use of spatial information of its neighboring region. The main objective of in-painting is reinstallation of damaged pixel value and elimination of selected object from image. Image in-painting technique is used to remove scratches from old photographs. In each case, the goal is to fill the damaged portion of the image in a visually plausible way that is to produce a modified image in which the in-painted region is merged into the image so seamlessly that a typical viewer is not aware that any modification has occurred and the new image have close resemblance to the original image and we are not able to differentiate between original image and the in-painted image. we are trying to use opencv's twos technique of image in-painting they are cv.INPAINT_TELEA and cv.INPAINT_NS (Navier-Stokes).**

I. INTRODUCTION

Image in-painting is the process of filling in missing data in a designated region of a still Image. Applications range from removing objects from a scene to re-touching damaged paintings and photographs. The goal is to produce a revised image in which the in-painted region is seamlessly merged into the image in a way that is not detectable by a typical viewer. Traditionally, in-painting has been done by professional artists. For photography and film, in-painting is used to revert deterioration (e.g., cracks in photographs or scratches and dust spots in film), or to add or remove elements (e.g., removal of stamped date and redeye from photographs, the infamous "airbrushing"). We us Telea and Navier-Stokes algorithm for digital in-painting of still images. It involves a direct solution of the Navier-Stokes equations and Telea function.After the user selects the regions to be restored, the algorithm automatically transports information into the in-painting region. The fill-in is done in such a way that lines arriving at the region's boundaries are completed inside. The technique introduced here does not require the user to specify where the novel information comes from. This is done automatically (and in a fast way), thereby allowing for simultaneously fill-in of multiple regions containing completely different structures and surrounding backgrounds. In addition, no limitations are imposed on the topology of the region to be in-painted.Since our in-painting algorithm is designed for removal of undesired objects on the image, the regions to be in-painted is generated using color detecting algorithm in which we detect black and white mask and then bitwise and with the original image then we get the color mask which is used to restore the image using given algorithms.

(a)Motivation

In image restoration, the image is restored closed to its original form. Sometimes the image is damaged or colored in some parts. This type of images are hard to view, understand and process for further use. Image in-painting is a technique which removes this distortions in image but the only problem with technique is that the user needs to define or state the mask beforehand ,only then it will minimize the distortion in image.we tried to automate this process using custom color detection algorithm which is use to get the mask which in turn can be used to image restoration

II. LITERATURE REVIEW

First note that image restoration is to filling-in, since the regions of missing data are usually large. That is, regions occupied by top to bottom scratches along several film frames, long cracks in photographs and so on, are of significantly larger

In [1] The authors set up a partial differential equation (PDE) to update image intensities inside the region with the above constraints.

The image smoothness information is estimated by the image Laplacian and it is propagated along the isophotes (contours of equal intensities). The isophotes are estimated by the image gradient rotated by 90 degrees.The authors show that these equations are closely related in form to the Navier-Stokes equations for 2D incompressible fluids .The benefit of reducing the problem to one of fluid dynamics is that we benefit from well developed theoretical analysis and numerical tools

In [2] Methods implementation solves the same constraints using a different technique. Instead of using the image Laplacian as the estimator of smoothness, the author uses a weighted average over a known image neighborhood of the pixel to inpaint. The known neighborhood pixels and gradients are used to estimate the color of the pixel to be inpainted.

Once a pixel is inpainted, the boundary needs to updated. The author treats the missing region of the image as level sets and uses the fast matching method to update the boundary.

III. Methodology

To explain our method, consider Figure 2, in which one must in-paint the point p situated on the boundary $\partial\Omega$ of the region to in-paint Ω. Take a small neighbourhood $B\varepsilon(p)$ of sizeε of the known image around p (Figure 2). the in-painting of p should be determined by the values of the known image points close We first consider gray value images, color images being a natural extension . For ε small enough, we consider a first order approximation $Iq(p)$ of the image in point p, given the image $I(q)$ and gradient $\nabla I(q)$ values of point q (Figure 2(b)):

$$Iq(p)=I(q)+\nabla I(q)(p-q).$$

(a) Mathematical model

This section describes our in-painting method. First, we introduce the mathematical model on which we base our in-painting (Section a). Next, we describe how the missing regions are in-painted using the FMM (Section b). Finally, we detail the implementation of in-painting one point on the missing region's boundary (Section c).

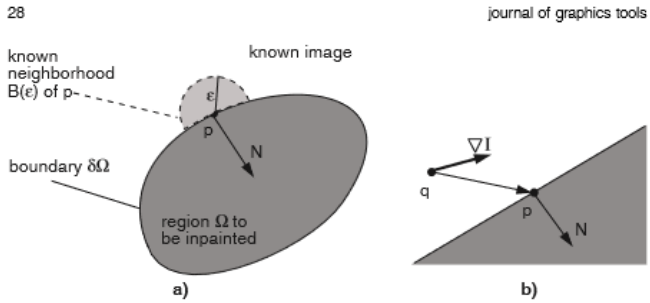


Figure 2. The inpainting principle.

Next, we inpaint point p as a function of all points q in $B\varepsilon(p)$ by summing the estimates of all points q, weighted by a normalized weighting function w(p, q):

$$I(p) = \sum_{q\in B\varepsilon(p)} w(p,q)[I(q) + \nabla I(q)(p-q)] \Big/ \sum_{q\in B\varepsilon(p)} w(p,q)$$

The weighting function w(p, q), detailed in Section 2.3, is designed such that the in-painting of p propagates the gray value as well as the sharp details of the image over $B\varepsilon(p)$.

(b) Adding In-painting to the FMM Section

To inpaint the whole Ω, we iteratively apply Equation 2 to all the discrete pixels of $\partial\Omega$, in increasing distance from $\partial\Omega$'s initial position $\partial\Omega i$, and advance the boundary inside Ω until the whole region has been inpainted (see pseudocode in Figure 3). In-painting points in increasing distance order from $\partial\Omega i$ ensures that areas closest to known image points are filled in first, thus mimicking manual in-painting techniques [Bertalmio 00, Bertalmio 01]. Implementing the above requires a method that propagates $\partial\Omega$ into Ω by advancing the pixels of $\partial\Omega$ in order of their distance to the initial boundary $\partial\Omega i$. For this, we use the fast marching method. In brief, the FMM is an algorithm that solves the Eikonal equation: $|\nabla T| = 1$ on Ω, with $T = 0$ on $\partial\Omega$. (3) The solution T of Equation 3 is the distance map of the Ω pixels to the boundary $\partial\Omega$. The level sets, or isolines, of T are exactly the successive

boundaries $\partial\Omega$ of the shrinking Ω that we need for in-painting. The normal N to $\partial\Omega$, also needed for in-painting, is exactly ∇T. The FMM guarantees that pixels of $\partial\Omega$ are always processed in increasing order of their distanceto-boundary

. The FMM maintains a so-called narrow band of pixels, which is exactly our in-painting boundary $\partial\Omega$. For every image pixel, we store its value T, its image gray value I (both represented as floating-point values), and a flag f that may have three values:

- BAND: the pixel belongs to the narrow band. Its T value undergoes update.
- KNOWN : the pixel is outside $\partial\Omega$, in the known image area. Its T and I values are known.
- INSIDE: the pixel is inside $\partial\Omega$, in the region to inpaint. Its T and I values are not yet known. The FMM has an initialization and propagation phase as follows. First, we set T to zero on and outside the boundary $\partial\Omega$ of the region to inpaint and to some large value inside, and initialize f over the whole image as explained above. All BAND points are inserted in a heap

(a) In-painting One Point

We consider now how to inpaint a newly discovered point (k,l), as a function of the KNOWN points around it. We iterate over the KNOWN points in the neighborhood $B\varepsilon$ of the current point (i, j) and compute I(i, j) following Equation 2. The image gradient ∇I is estimated by central differences. close to the FMM's information propagation direction, is higher than for those farther from N. The geometric distance component dst(p, q) decreases the contribution of the pixels geometrically farther from p. The level set distance component lev(p, q) ensures that pixels close to the contour through p contribute more than farther pixels. Both dst and lev are relative with respect to the reference distances d0 and T0. In practice, we set d0 and T0 to the interpixel distance, i.e., to 1. Overall, the above factors model the manual in-painting heuristics that describe how to paint a point by strokes bringing color from a small region around it. For ε up to about six pixels, i.e., when in-painting thin regions, dst and lev have a weak effect. For thicker regions to inpaint, such as Figure 8(d), where we used an ε of 12 pixels, using dst and lev provides better results than using dir alone. The above is clearly visible in Figure 6, on a test image taken from [Bertalmio 00], where the missing ring-shaped region is more than 30 pixels thick. Figure 6(c) shows, on an image detail, the effect of dir alone. The results are somewhat less blurry when dir and dst (Figure 6(d)) or dir and lev are used

together. The in-painting is the best visually when all three components are used

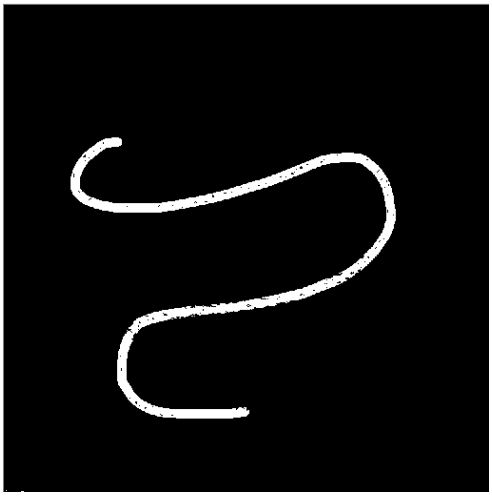
V. EXPERIMENTATION AND RESULT ANALYSIS

We have taken an image which has less blue backgroung to optimize the result and drawn an distortion with paint with blue color.

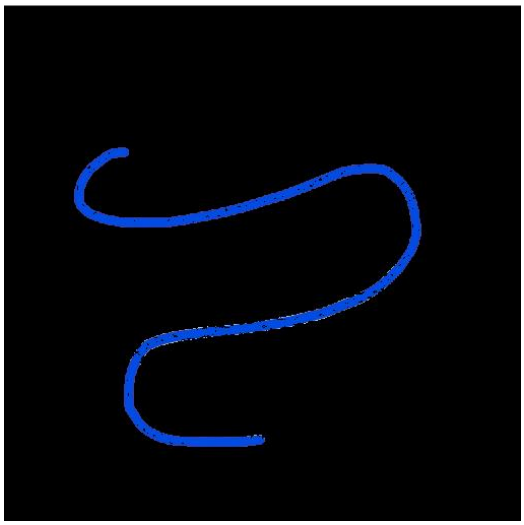
A Original Image

Now we extract mask in black and white format buy using color detection algorithm.In this we first convert image in hsv format and then extract mask buy specifying the upper and lower bound of the color blue.

B Mask

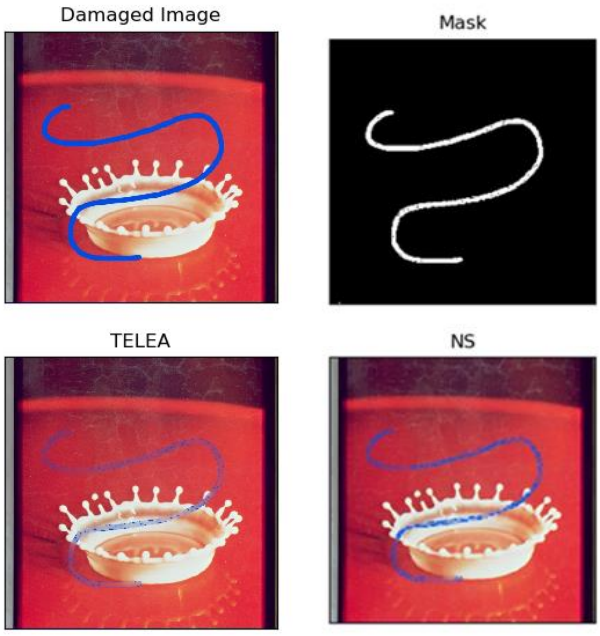


Then to get a color mask we bitwise and the mask and the original image to get the colored mask



C Bitwise and mask with original image

Then we use opencv (telea and ns) to remove the distortion as much as we can and get the optimal output.



V. Conclusion

In this report ,we present a simple method for image restoration for a single color using in-painting technique. For future work, I would like to generalize the process using ANN which can detect any color and any distortation. For this project, the main media files were images which were distorted from start. The image to be restored is distorted buy ourself with blue color .

References

[1] M. Bertalm, A.l.bertozzi , G.sapero., " Navier-Stokes, Fluid Dynamics, and Image and Video In-painting," ,unpublished

[2] Telea, Alexandru., "An Image In-painting Technique Based on the Fast Marching Method," ,unpublished

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