4-21-2023

Image Inpainting Methods: Digital Image Reconstruction and Restoration

Haruka Kido
haruka.kido@und.edu

Image Inpainting Methods: Digital Image Reconstruction and Restoration

Haruka Kido, *Electrical Engineering, University of North Dakota*

***Abstract*—This report investigates the digital image processing of image inpainting methods, particularly for digital image reconstruction and restoration through the two computational tools grouped into: [1] MatLAB/Image Segmenter and [2] Anaconda/OpenCV/Python. The use cases explored in the project involve image reconstruction and restoration of celestial imageries for means of clear demonstration and subject-matter consistency but can extend to more artistic purposes involving the removal of unwanted objects within the backgrounds or foregrounds of images that can be "erased" or "hidden" by being replaced by neighboring pixels of similar characteristics for image reconstruction and the removal of damaged parts observed in old photographs damaged by noises, dark streaks, faded or scratched edges, folds, physio-chemical alterations, ink blotches, or technological obscurities (such as lens flare, lens aberrations, or crop marks) for image restoration. The celestial images used for the purposes of this project are taken from the public collection of NASA's James Webb and Hubble Telescope image archives.**

***Index Terms*—digital image processing, image inpainting methods, digital image reconstruction, digital image restoration, MatLAB, Image Segmenter App, Graph Cut, ROI, binary mask, Anaconda, OpenCV, Python, celestial imagery, telescope images, astrophotography, micrometeoroid mirror damage**

I. Introduction

THIS REPORT is a technical paper for the Final Term Project required by EE456: Digital Image Processing. The objective of the project is to research and develop the algorithmic processes of digital image inpainting methods. The two selected tools are: [1] MatLAB/Image Segmenter and [2] Anaconda/OpenCV/Python. The latter relies on the generation of image inputs as binary masks (overlayed over the original image) built from the Image Segmenter within MatLAB. The report demonstrates the theories behind image reconstruction and restoration methods through mathematical explanations, respective algorithms, computational environments, manual processes, image inputs and outputs, and intermediary segmentations. Astrophotography images with undesirable background objects for purposes of digital image reconstruction and with interest point marks or damages for digital image restoration are used for case studies in the implementation of digital image inpainting algorithms. Each part of this report is subdivided into the following image inpainting techniques: A. *Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method (MatLAB)*, B. *Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)*, C. *Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm (Anaconda/OpenCV/Python)*, and D. *Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)*. These methods are grouped into 2 categories of image inpainting algorithm techniques: [1] exemplar-based inpainting and [2] partial-differential equations (PDE)-based inpainting. In this paper, the techniques explored with the pure use of MatLAB/Image Segmenter deliver inpainted results through exemplar-based inpainting while the techniques explored in the Anaconda/OpenCV/Python tool deliver inpainted results (while more susceptible to blurring due to diffusion) through PDE-based inpainting. *Note*: In some cases, the generated binary mask has undergone inversion during the post-segmentation phase.

II. Related Works and Theoretical Foundations

A. *Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method (MatLAB)*

Image inpainting technique can be used for digital image reconstruction purposes, whereby images with unwanted background objects (while not actually classified as damaged) can be rebuilt after the replacement of undesirable background pixels with neighboring pixels of similar intensity, threshold, color combination, structure, or topology. The Coordinate-Planed Ellipse Method relies on the exemplar-based image inpainting function, a widely known technique for object removal through sampling and copying of texture and color values from known parts of the image across the boundary of the region to be inpainted [1]:

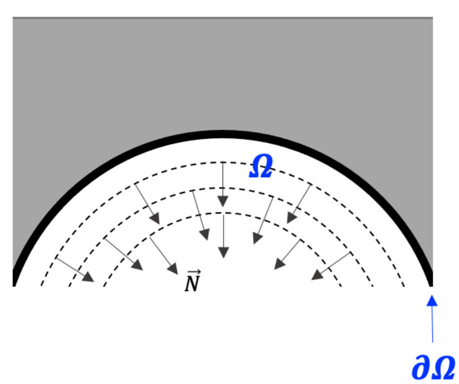


Figure 1. Pixel propagation across boundary of the region to be inpainted as normal to the isophote lines

When the binary masks are built, the background pixels surrounding the target region (foreground) are sampled and patch-priority maximization enables finding the exemplar that minimizes the error of priority computation. Figure 2 abstracts this exemplar-based algorithmic approach at the "fill front" where the region-filling algorithm implements top-patch matching and copying for texture synthesis [3]:

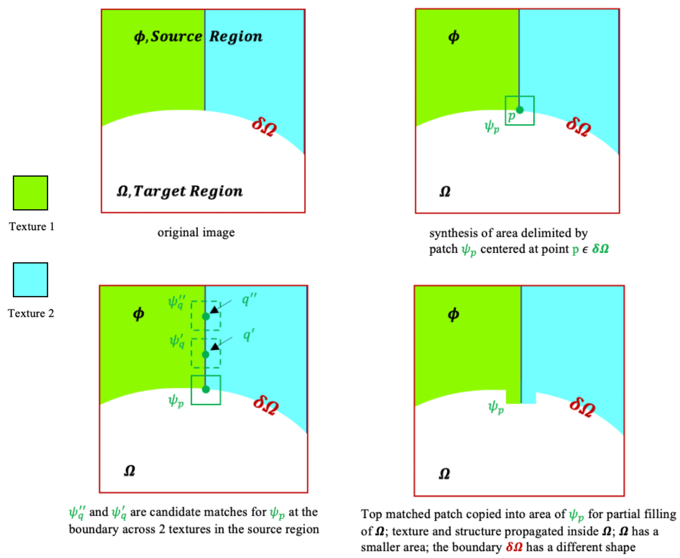
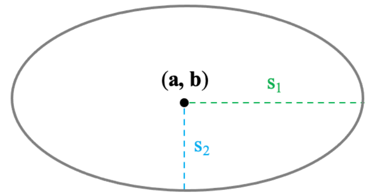


Figure 2. Patch-prioritization for exemplar-based texture and structure propagation at the boundary between source region and target region

Mathworks develops the process to create a mask using the properties of an ellipse [2]; This editing effect is accomplished in MatLAB by the application of a binary ROI mask applied to a drawn ellipse ROI function that is ascertained by its center point (x, y) and changes in the x-semiaxis and y-semiaxis coordinate points.

(x, y): ellipse center point
s_1: x-semiaxis
s_2: y-semiaxis



$$\frac{(x-\mathbf{a})^2}{s_1{}^2} + \frac{(y-\mathbf{b})^2}{s_2{}^2} = 1$$

$$\frac{x^2}{s_1{}^2} + \frac{y^2}{s_2{}^2} = 1$$

Figure 3. Theory of Ellipses

B. *Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)*

The Graph Cut Method also relies on exemplar-based inpainting but enables manual delineation of the boundary of the region to be inpainted by demarcation of the background and foreground on the original image, resulting in segmentation automation. The process is hence considered semiautomatic and flexible to pre-segmentation refinements. Graph Theory is applied to image processing by partitioning each pixel as a node connected by hierarchical weighted edges, with increasing weight corresponding to increasing probability of pixel relation [7]. The graph cut segmentation is implemented by cutting along the weak edges.

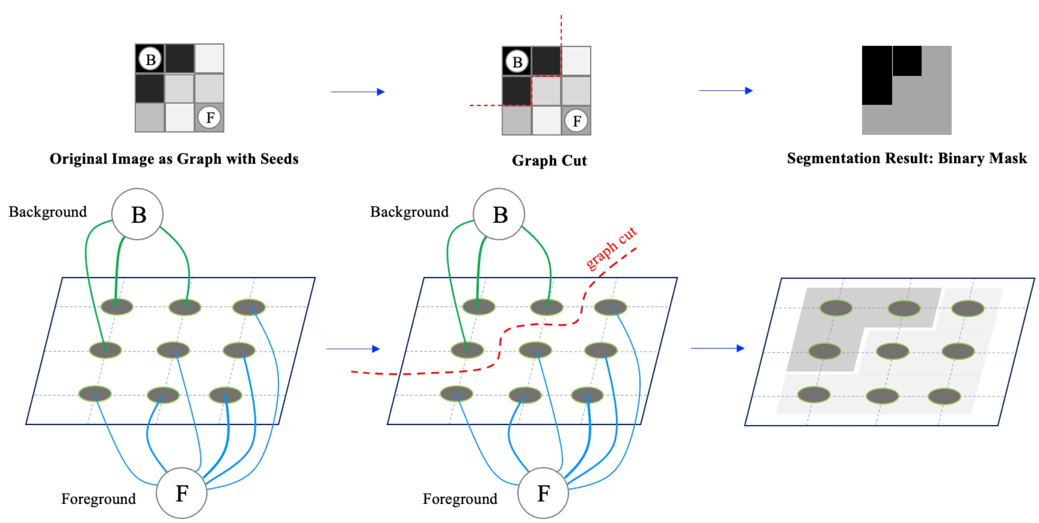


Figure 4. Nodal partitioning in the Graph Cut Method: Segmentation by Graph Cut Method

C. *Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm (Anaconda/OpenCV/Python)*

Digital image restoration is used to repair damaged parts of images through the Fast-Marching Method (FMM) as Telea's Algorithm. The Fast-Marching Method (FMM) as Telea's Algorithm enables the replacement of high-gradient image regions by the following: determination of the filling order of pixels to be inpainted and use of a weighted first-order Taylor series approximation from a neighborhood of pixels, ε, surrounding the selected regions to be inpainted [11].

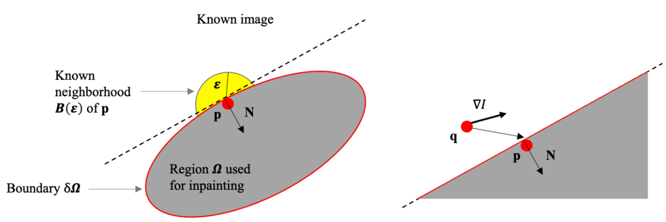


Figure 5. Image Inpainting Theory

The user-defined ε-neighborhood size of the neighboring region around the target region sets the boundary around the region to be inpainted. Starting with Image Inpainting Theory, the Fast-Marching Method (FMM) relies on the presumption that a small enough ε is realized; The mathematical model with the first-order approximation $I_q(p)$ of the image in point p, from the image $I(q)$ and gradient $\nabla I(q)$ values of point q is built from the equation:

$$I_q(p) = I(q) + \nabla I(q)(p - q) \qquad [1]$$

The inpaint of point p is a function of all points q in $B_\varepsilon(p)$ through summing of the estimates of all points q,weighted by the normalized weighting function $\boldsymbol{w(p,\ q)}$, as described in Equation [2]:

$$I(p) = \frac{\sum_{q \epsilon B_\varepsilon(p)} \boldsymbol{w(p, q)}[I(q) + \nabla I(q)(p - q)]}{\sum_{q \epsilon B_\varepsilon(p)} \boldsymbol{w(p, q)}} \qquad [2]$$

The normalized weight function is applied with p as the pixel to be inpainted and q as the pixel in p's neighborhood:

$$\boldsymbol{w(p,\ q)} = dir(p,\ q) \cdot dst(p,\ q) \cdot lev(p,\ q) \qquad [3]$$

where $dir(p,\ q)$ refers to the directional component, $dst(p,\ q)$ refers to the distance component, and $lev(p,\ q)$ refers to the level set component. The directional component, $dir(p,\ q)$, depends on

$$N = \nabla T \qquad [4]$$

where T is the level set (time of arrival of front) function obtained by solving the Eikonal Equation and ∇ is the gradient operator. The first-order Fast-Marching Method as Telea Algorithm is essentially a method of diffusion (or propagation of inpainting) using the gradient operator demonstrated in smoothing the field T (at a pixel, as either $-T_{out}$ or T_{in}) using the filter and computing the ∇T using central differences. It is traditionally known as a solution to the Eikonal Equation.

According to Huan, Murali, and Ali, the authors of "Image restoration based on the fast-marching method and block-based sampling," Telea states ''*The main limitation of our method (. . .) is the blurring produced when inpainting regions thicker than 10–15 pixels, especially visible when sharp isophotes intersect the region's boundary almost tangentially*. [10]" It is widely researched that the inpainting with FMM can be done iteratively with Equation [2] until the entire region Ω is inpainted, a computationally lengthy process if the Anaconda/OpenCV/Python environment is used. As this first-order FMM method applied on thicker target regions produces blurred results from the diffusion process, the SSD (Sum of Squared Differences) method continues to be researched as a more robust second-order FMM method.

D. *Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)*

Another common method for digital image restoration is the Navier-Stokes Method. The Navier-Stokes Method is researched to be based on image inpainting on isophotes and uses ideas from classical fluid dynamics. The image intensity is analyzed as a "stream function" for a two-dimensional incompressible fluid flow. The image intensity's Laplacian is used in the vorticity of the fluid and is transported to the region of interest to be inpainted through a vector field defined by the stream function. The NS algorithm is known to result in the continuation of isophotes and match gradient vectors at the boundary of the inpainting region [12].

Like the FMM discussed previously, lower-order partial differential equations-based methods of image inpainting give image quality problems due to the typical enforcing of only one of the boundary conditions from that of the Dirichlet (fixed I) and the boundary direction of ∇I, resulting in an inpainted image with discontinuities in the slope of isophote lines or an I differential along the boundary. The vector evolution for ∇I makes the Navier-Stokes Method a more robust method of image inpainting due to its ability to manifest the continuity of the image intensity function I and the propagation of its isophote direction across the boundary between the inpainted region and surrounding neighborhood pixel region. Analogously, the Navier-Stokes Method causes the image intensities I_n to "diffuse" in continuum in fluidity across the boundary between the region to be inpainted and the surrounding exterior region while the isophote directions propagate across it; The result is a smoothness of image intensity in the direction of the isophotes from gradient-matching, as shown below in Figure 6,

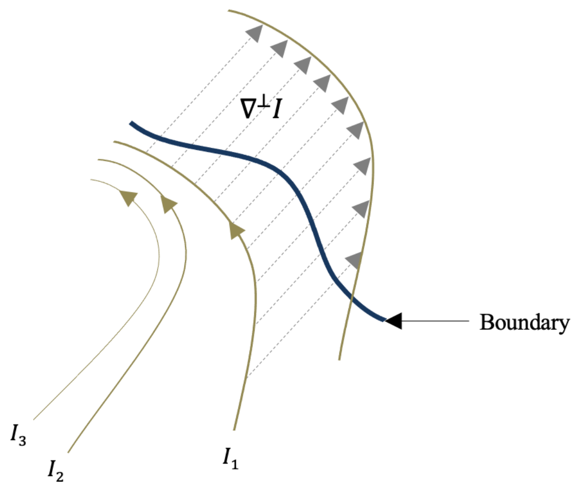


Figure 6. Image Inpainting Along Isophotes

where I is the image intensity, $\nabla^{\perp} I$ is the isophote direction, $\nabla^{\perp}$ is the perpendicular gradient, ΔI is the smoothness, Δ is the Laplacian operator $\partial_x^2 + \partial_y^2$, and v is the anisotropic diffusion across the boundary, given the discrete approximation of the partial differential equation:

$$I_t = \nabla^{\perp} I \cdot \nabla \Delta I \qquad [5]$$

Equation [6] shows the evolution of Equation [5] into a steady-state solution to satisfy the condition that the isophote direction is kept parallel to the level curves of the smoothness of the image intensity, causing the anisotropic diffusion to equal 0:

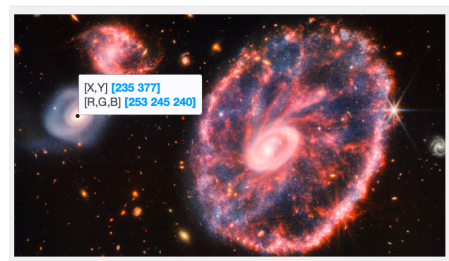
$$\nabla^{\perp} I \cdot \nabla \Delta I = 0. \qquad [6]$$

III. Methodology

The methodologies of the image inpainting techniques include the diagrammatic computational processes that apply the system models:

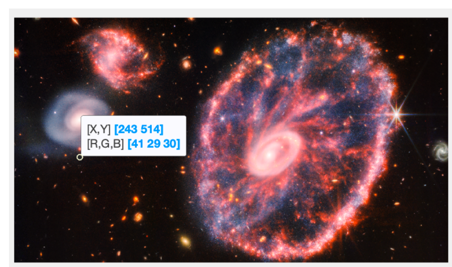
A. *Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method (MatLAB)*

The MatLAB Algorithm for Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method is run on an image of the Cartwheel Galaxy (NIRCam and MIRI Composite Image) from the Webb Space Telescope by NASA. The main object of interest is taken as the larger cartwheel galaxy composed of more pixels than the peripheral galaxy that is eliminated by its ellipse ROI delineation. The ellipse location identification is possible through point markings of the ellipse's center point, change in x-semiaxis, and change in y-semiaxis; The result of the ellipse location identification is the binary ROI mask, which is applied over the original image to obtain the inpainted image through exemplar-based image inpainting. The white region of the mask ascertains the region of interest around which the surrounding neighboring pixels are taken as the replacement pixels.



center point

change in x-semiaxis



change in y-semiaxis

Figure 7. Point Markings from Data Tips

B. *Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)*

Image segmentation and graph cut method is used for image reconstruction by the manual demarcation of the foreground and background areas of a loaded original image in MatLAB's Image Segmenter. MatLAB builds a binary mask and loads an image segmenter function that stores the pixel locations of the foreground (white) and background (black). The image segmenter program converts the RGB image into L*a*b color space and uses a superpixel configuration to the arguments of the foreground and background. The source code must then reference the image segmenter function for the input of the binary masked image over the original image to be used for the subsequent inpaint function. Similar to the Coordinate-Plane Ellipse ROI Method, the white part of the binary mask marks the region of interest as the enclosed boundary disclosed from the surrounding neighboring pixels as the replacement group of topological similarity. This binary mask is overlayed the original image through an inpaint function in MatLAB (“`InpaintExemplar()`”).

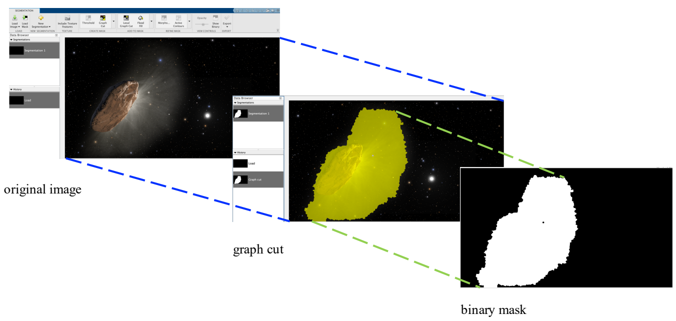


Figure 8. Process for Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)

C. *Digital Image Restoration: Fast-Marching Method (FMM) as Telea’s Algorithm (Anaconda/OpenCV/Python)*

Computationally, image segmentation through the Fast-Marching Method (FMM) produces inpainted images from a “flood fill” segmentation used to iteratively smooth damaged topographic surfaces through an overlay of an inverted binary mask. The resulting repaired parts, if large enough, constitute regions comparable to watershed segmentation environments with catchment basins.

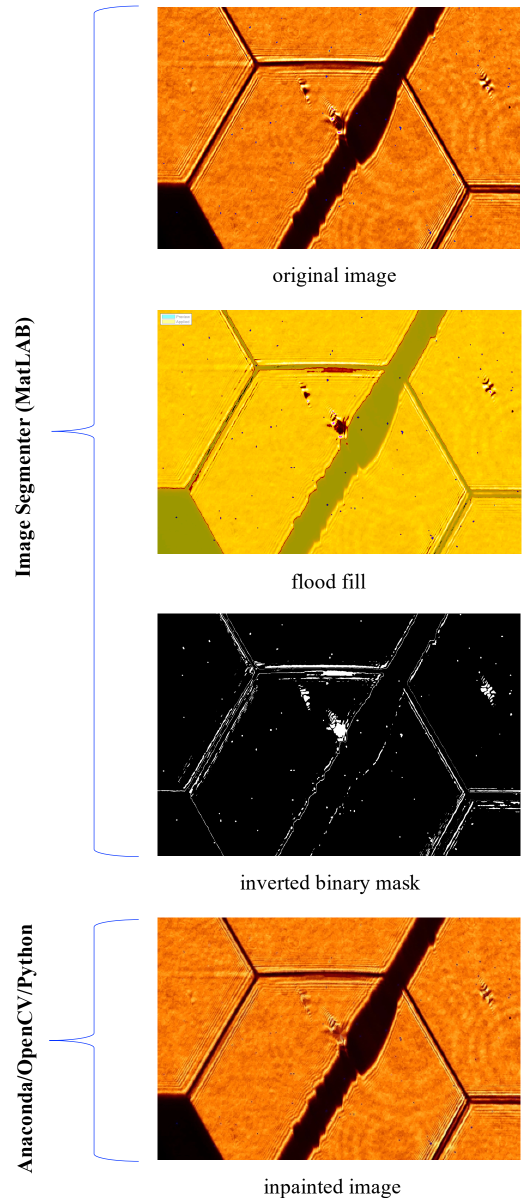
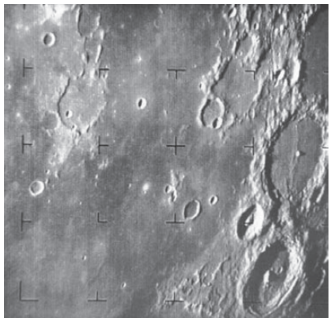


Figure 9. Process Images for Digital Image Restoration: Fast-Marching Method (FMM) as Telea’s Algorithm (MatLAB/Image Segmenter) using the image “*A Micrometeroid Damaged a Mirror of the Webb Telescope*” by NASA

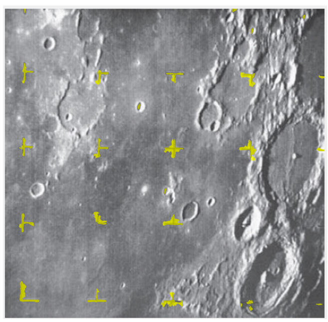
D. *Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)*

Navier-Stokes Method includes a high-precision graph cut binary mask to implement structure and color propagation across the boundary between the target and source regions:

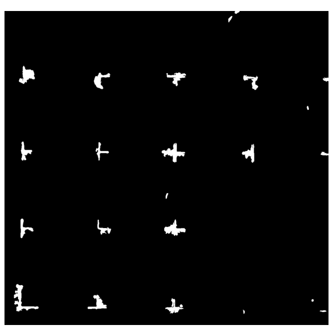
Image Segmenter (MatLAB)



original image



graph cut



binary mask

Anaconda/OpenCV/Python

inpainted image

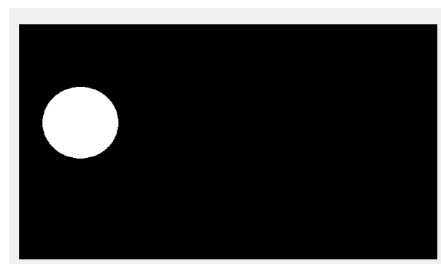
Figure 10. Process Images for Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)

IV. Experimental Results

A. *Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method (MatLAB)*

In this case study using the Cartwheel Galaxy image, the MatLAB algorithm outputs effectively demonstrate the process of background object elimination as image reconstruction.

ellipse location identification



binary ROI mask

inpainted image

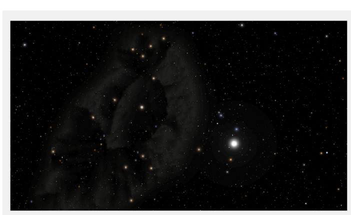
Figure 11. MatLAB Algorithm Outputs for Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method

B. *Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)*

The inpainted image is a reconstructed image with the graph cut foreground of the binary mask utilized to eliminate the undesired object. In this case, the purpose is the keep the background topology of the star-light night sky by replicating its consistent topological surface. While there is a trace of a phantom residue of the eliminated object, the algorithm works adequately well for the purposes of image reconstruction. The respective MatLAB Algorithm Outputs are shown below:

original image

binary mask

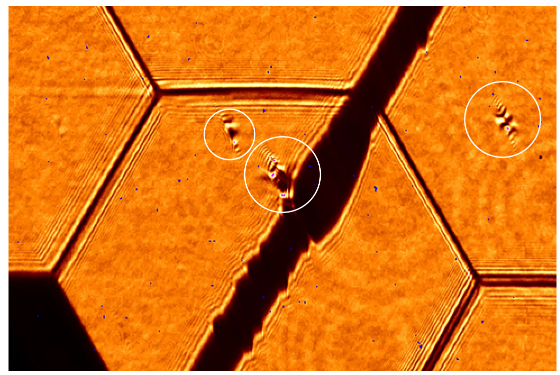


inpainted image

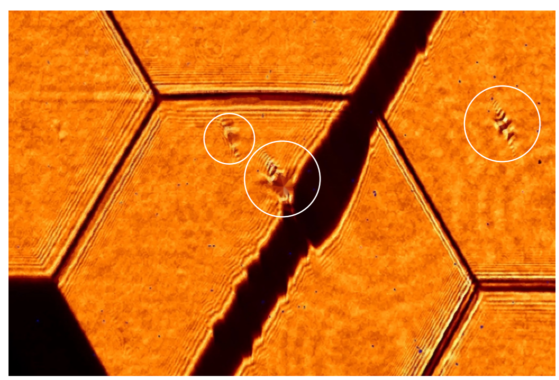
Figure 12. MatLAB Algorithm Outputs for Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)

C. *Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm (Anaconda/OpenCV/Python)*

From the methodology, the inpainted image as the output is investigated further for the experimental analysis. A closer comparison between the original image and the inpainted image shows the differences in the grouped pixels that correspond to the Webb telescope mirror damages from a micrometeoroid:



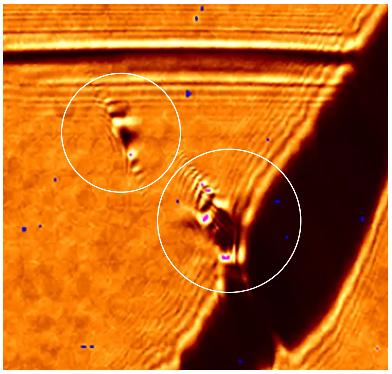
original image



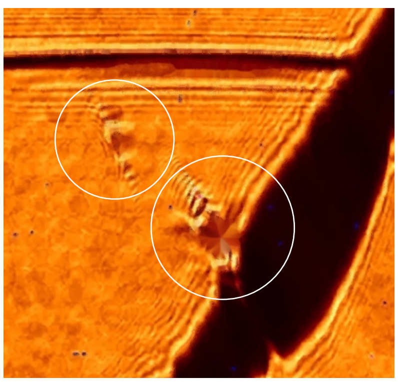
inpainted image

Figure 13. Original Image and Inpainted Image from Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm

While the contour edges of the honey combed shaped mirror geometries are restored without obstructive blurring, the blurring produced by the FFM Telea Algorithm is evident for the thickest region that the algorithm is applied to, as shown most closely in the inpainted image of Figure 14:



original image



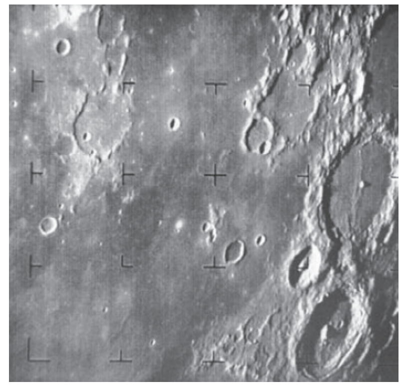
inpainted image

Figure 14. Original Image and Inpainted Image from Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm, Zoomed-In Version

The finely-detailed blue-toned spots shown across the original image frame resemble ink blotches as errors to be eliminated and are indeed eliminated after the application of Telea's algorithm.

D. *Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)*

The inpainted output is cleared of the undesired interest point markers. Faint residues of the interest point markers can only be observed at close ranges to the maximum zoom.



original image

inpainted image

Figure 15. Original Image and Inpainted Image from Digital Image Restoration: Navier-Stokes Method

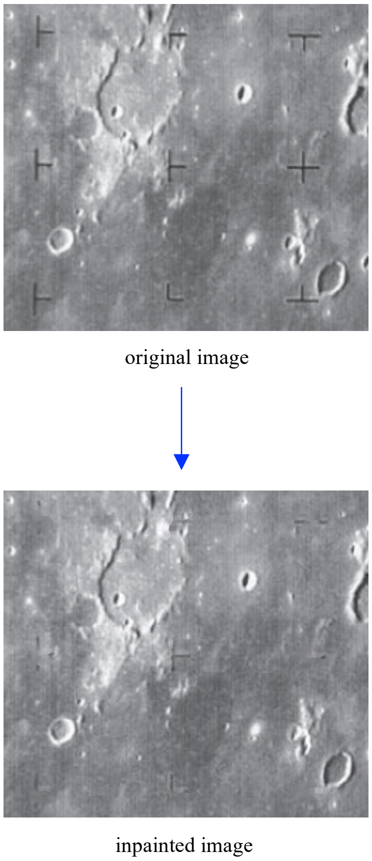


Figure 16. Original Image and Inpainted Image from Digital Image Restoration: Navier-Stokes Method, Zoomed-In Version

V. Source Codes

A. *Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method (MatLAB)*

```
clear;
original_image = imread('Galaxy.png');
subplot(1, 3, 1);
imshow(original_image);

%draw ellipse as ROI
ellipse = drawellipse('Center', [235 377], 'SemiAxes', [145 137]);
mask = createMask(ellipse); %make mask for ROI as ellipse
subplot(1, 3, 2);
imshow(mask);

inpainted_image = inpaintExemplar(original_image, mask);
subplot(1, 3, 3);
imshow(inpainted_image);
```

Figure 17. MatLAB Algorithm for Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method

B. *Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)*

The segment image function algorithm generated by the Image Segmenter and the source code referencing the masked image from the segment image function algorithm to implement the image segmentation and graph cut method are shown sequentially below:

```
ImageInpainting2.m   segmentImage.m
function [BW,maskedImage] = segmentImage(RGB)
%segmentImage Segment image using auto-generated code from imageSegmenter app
%  [BW,MASKEDIMAGE] = segmentImage(RGB) segments image RGB using
%  auto-generated code from the imageSegmenter app. The final segmentation
%  is returned in BW, and a masked image is returned in MASKEDIMAGE.

% Auto-generated by imageSegmenter app on 10-Apr-2023
%----------------------------------------------------


% Convert RGB image into L*a*b* color space.
X = rgb2lab(RGB);

% Graph cut
foregroundInd = [82013 82022 82727 82744 84189 84193 84882 85598 85638 85642 87807 87808 88473 89251      504356 ];
backgroundInd = [2982 2983 3524 3652 3661 3670 3675 3680 3689 3699 3704 3717 3741 3762 3769 3790 3822      919670 ];
L = superpixels(X,4708,'IsInputLab',true);

% Convert L*a*b* range to [0 1]
scaledX = prepLab(X);
BW = lazysnapping(scaledX,L,foregroundInd,backgroundInd);

% Create masked image.
maskedImage = RGB;
maskedImage(repmat(~BW,[1 1 3])) = 0;
end

function out = prepLab(in)

% Convert L*a*b* image to range [0,1]
out = in;
out(:,:,1) = in(:,:,1) / 100;  % L range is [0 100].
out(:,:,2) = (in(:,:,2) + 86.1827) / 184.4170;  % a* range is [-86.1827,98.2343].
out(:,:,3) = (in(:,:,3) + 107.8602) / 202.3382;  % b* range is [-107.8602,94.4780].

end
```

Figure 18. Segment Image Algorithm for Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)

```
ImageInpainting2.m   segmentImage.m
clear;
RGB = imread('Oumuamua.png');
subplot(1, 3, 1);
imshow(RGB);

[BW,maskedImage] = segmentImage(RGB);
subplot(1, 3, 2);
imshow(BW);

inpainted_image = inpaintExemplar(RGB, BW);
subplot(1, 3, 3);
imshow(inpainted_image);
```

Figure 19. Image Inpainting Algorithm for Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)

C. *Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm (Anaconda/OpenCV/Python)*

Anacoda is installed to activate the OpenCV environment for Python to be used as a programming language for reading in the original image, inpainting the original image with a binary mask image developed in MatLAB, and ultimately writing out the inpainted image in the same working directory.

```
Last login: Tue Apr  4 14:49:59 on ttys000
(base) harukakido@Harukas-MBP ~ % conda activate opencv-env
(opencv-env) harukakido@Harukas-MBP ~ % conda install -c conda-forge opencv
Collecting package metadata (current_repodata.json): done
Solving environment: done
# All requested packages already installed.
(opencv-env) harukakido@Harukas-MBP ~ % python
Python 3.11.2 | packaged by conda-forge | (main, Mar 31 2023, 17:54:27) [Clang
14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import cv2
>>> original = cv2.imread('/Users/harukakido/Webb_Telescope_Damaged.jpg')
>>> mask = cv2.imread('/Users/harukakido/binarymask.jpg', 0)
>>> inpainted = cv2.inpaint(original, mask, 3, cv2.INPAINT_TELEA)
>>> cv2.imwrite('Inpainted_Webb_Telescope_Damaged', inpainted)
True
```

Figure 20. Anaconda/OpenCV/Python Algorithm in the Desktop Terminal for Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm (MatLAB/Image Segmenter)

D. *Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)*

To implement the Navier-Stokes Method as the image inpainting technique, the Anaconda/OpenCV/Python environment is used to apply the binary mask built in Image Segmenter over the original image in the argument for the inpaint function:

```
Last login: Tue Apr  4 14:38:53 on ttys000
(base) harukakido@Harukas-MBP ~ % conda activate opencv-env
(opencv-env) harukakido@Harukas-MBP ~ % conda install -c conda-forge opencv
Collecting package metadata (current_repodata.json): done
Solving environment: done
# All requested packages are already installed.
(opencv-env) harukakido@Harukas-MBP ~ % python
Python 3.11.2 | packaged by conda-forge | (main, Mar 31 2023, 17:54:27) [Clang
14.0.6 ] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import cv2
>>> original = cv2.imread('/Users/harukakido/FirstMoonImage_Ranger.jpg')
>>> mask = cv2.imread('/Users/harukakido/binarymask.jpg', 0)
>>> inpainted = cv2.inpaint(original, mask, 3, cv2.INPAINT_NS)
>>> cv2.imwrite('Inpainted_FirstMoonImage_Ranger.jpg', inpainted)
True
>>>
```

Figure 21. Anaconda/OpenCV/Python Algorithm in the Desktop Terminal for Digital Image Restoration: Navier-Stokes Method (MatLAB/Image Segmenter)

VI. Discussion of Results

A. *Digital Image Reconstruction: Coordinate-Plane Ellipse ROI Method (MatLAB)*

The Coordinate-Plane Ellipse ROI Method for purposes of image reconstruction works both effectively and quickly with the accurate application of the parameters of an ellipse. When objects of celestial image backgrounds are segmented using value sets obtained through the Point Marks function from MatLAB's Data Tips, the generated binary ROI mask is not considered manually based. The segmentation of common geometries in ROI methods makes it easier to discern and reapply the outlines of objects and the boundary difference between the foreground and the background; Hence the inpainted image output can produce perceptually higher-quality results. Color-redundancy in hue, saturation, and value can be beneficial when as a property of the background of an image to be inpainted.

B. *Digital Image Reconstruction: Image Segmentation and Graph Cut Method (MatLAB/Image Segmenter)*

The Graph Cut Method can be valuable for image inpainting requirements of basic geometries of objects or of easily traceable areas to be segmented in the binary mask. The larger the patch size, the higher the filling rate in exemplar-based patch-filling, resulting in a faster computation, and making the Graph Cut method, like the Coordinate-Plane Ellipse ROI Method, robust enough for digital images that have distinctively large objects to be removed. In the celestial imagery examples applied in this paper, both the Coordinate-Plane Ellipse ROI Method and the Graph Cut Method produce significantly valuable inpainting results with minimal texture and color distortion since homogenous textures with repetitions of structure are present.

C. *Digital Image Restoration: Fast-Marching Method (FMM) as Telea's Algorithm (Anaconda/OpenCV/Python)*

The repaired groups of pixels clearly identified are the specs of physical matter (perhaps dust collections on the mirror) that cause higher-intensity image gradients in clusters that otherwise would not be present. These higher-intensity image gradients are also apparent in the contour delineations of the honey-combed shapes. The results of the experiment using NASA's image of the damaged Webb telescope mirror shows an effective smoothing effect of damaged group of pixels.

D. *Digital Image Restoration: Navier-Stokes Method (Anaconda/OpenCV/Python)*

It may be difficult to tell how "fluid" the marked regions to be inpainted are, due to the abstractions of these marked regions under the graph cut technique as linear marks at the interest point lines spread across the First Moon Image. The results show an efficient image restoration technique whereby the identified segmentation through graph cut is eliminated by the application of a binary mask in the Navier-Stokes Method. In this case, the interest point marks (perpendicular coordinate lines spread at an equidistance across the First Moon Image) are regarded as the "damaged" parts, eliminated for the repair apparent in the inpainted image.

VII. Conclusion and Future Work

This report is based on theoretical research, algorithm development, the computational process builds, and experimental work incorporating image inputs and outputs under the image inpainting techniques of image reconstruction and restoration. The author's contribution to the field of image inpainting technique consists of the analyses of the theoretical foundations of the image inpainting techniques in conjunction with the segmentation functions, all theoretical figures as visual descriptors of 2D image inpainting methods, and applications of chosen images (with manual segmentation when necessary) used for purposes of image reconstruction and restoration through the image inpainting implementation examples, as the series of celestial images, and lastly, the analyses of the effectiveness of the resulting inpainted images in comparison to the original NASA images.

While digital image inpainting is computationally faster in MatLAB/Image Segmenter, the differences in the effectiveness between the 2 tools grouped together, namely MatLAB/Image Segmenter and Anaconda/OpenCV/Python, should be studied further. Manual segmentation can be difficult and time-consuming but useful for images with finer details that require correction at the micro-imaging scale. For micro-imaging, the flood fill technique during the segmentation process is observed to work most effectively.

Additional future investigations include: ROI specification of the circle, polyline, rectangle, and polygon, classification of pixels that are partially enclosed by the ROI, sum of squared differences (SSD) texture synthesis within exemplar-based image inpainting, exemplar-based image inpainting with stochastic or inhomogeneous textures, more robust algorithms of the Fast-Marching Method and the Navier-Stokes Method with higher-order partial differential equations, and the integration of digital image processing functions, specifically including built-in image inpainting functions, as software with digital image acquisition (camera) hardware.

References and Footnotes

References

[1] Bertalmio, Mercelo, Sapiro, Guillermo, Caselles, Vicent, Ballester, Coloma, "Image Inpainting," Proceedings of the 27th annual conference on Computer graphics and inteactive techniques, July 2000

[2] Mathworks, "createMask," Mathworks Support Documentation, https://www.mathworks.com/help/images/ref/images.roi.assistedfreehand.createmask.html

[3] Criminisi, A., P. Perez, K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *IEEE Transactions on Image Processing.* Vol. 13, No. 9, 2004, pp. 1200–1212

[4] Kale, Shilpa J., Gawande P.D., "Region Filling and Object Removal by Exemplar-Based Image Inpainting," International Journal of Advanced Engineering Research and Studies, E-ISSN2249–8974

[5] Le Meur, Olivier, Gautier, Josselin, Guillemot, Christine, "Examplar-Based Inpainting Based on Local Geometry," 18th IEEE International Conference on Image Processing, 2011, DOI: 10.1109/ICIP.2011.6116441

[6] Mathworks, "Segment Image Using Graph Cut in Image Segmenter," Mathworks Support Documentation, https://www.mathworks.com/help/images/segment-image-using-graph-cut.html

[7] Boykov, Yuri, Funka-Lea Gareth, "Graph Cuts and Efficient N-D Image Segmentation," International Journal of Computer Vision 70(2), 109-131, 2006. DOI: 10.1007/s11263-006-7934-5

[8] Boykov, Yuri, Veksler, Olga, Zabih, Ramin, "Fast Approximate Energy Minimization via Graph Cuts," IEEE Transactions on PAMI, vol. 23, no. 11, pp. 1222-1239

[9] Le Meur, O., M. Ebdelli, C. Guillemot, "Hierarchical Super-Resolution-Based-Inpainting," IEEE Transactions on Image Processing. Vol. 22, No. 10, 2013, pp. 3779–3790

[10] Huan, Xiaoli, Murali, Beddhu, Ali, Adel L., "Image restoration based on the fast-marching method and block-based sampling," Computer Vision and Image Understanding 114 (2010) 847–856

[11] Telea, Alexandru, "An Image Inpainting Technique Based on the Fast-Marching Method," Journal of Graphics Tools, 2004

[12] Bertalmio, M., Bertozzi, A. L., Sapiro, G., "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting," Proc. ICCV, 2001, pp. 1335–1362

[13] Gonzalez, Rafael C., Woods, Richard E., "Digital Image Processing, Fourth Edition" Pearson, 2018

[14] Gonzalez, Rafael C., Woods, Richard E., Eddins, Steven L., "Digital Image Processing Using MATLAB, 2nd Edition" Gatesmark Publishing, 2009

[15] The MathWorks, Inc., "Image Processing Toolbox User's Guide Version 3" 1993-2001

[16] Webb's First Images, Webb Space Telescope, NASA: https://webbtelescope.org/news/first-images/gallery

[17] Hubble Space Telescope Images, NASA: https://www.nasa.gov/mission_pages/hubble/multimedia/index.html

Haruka Kido received a Bachelor of Science (B.S.) in interdisciplinary studies (minor in architecture) from Cornell University, Ithaca, NY in 2017. She has recently worked as an Embedded Processing Intern for NASA and is currently studying towards earning a Bachelor of Science (B.S.) in electrical engineering (minor in mathematics) from the University of North Dakota.