IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 9, November 2014.
www.ijiset.com
ISSN 2348 – 7968

Detection and Removal of Cracks in Digitized Paintings via Digital Image Processing

Mohammed AbdALLA A. Elmaleeh[1], Nawafil Abdul Wahab Farajalla[2], Amin Babikr A/Nabi Mustafa[2]

[1] Computer Engineering Department, Faculty of Engineering, Alneelain University, Khartoum, Sudan

[2] Faculty of Mathematics and Computer Sciences, University of Gezira, Madani, Sudan

[2]Communication Engineering Department, Faculty of Engineering, Alneelain University, Khartoum, 12702, Sudan

Abstract

Many paintings, especially old ones, suffer from breaks in the substrate, the paint, or the varnish. These patterns are usually called cracks and be caused by aging, drying and mechanical factors. The appearance of cracks on painting deteriorates the perceived paintings quality. One can use digital image processing techniques to detect and eliminate the cracks on the digitized paintings. The main objective of this study is to present the digital image processing technique that can be applied to the virtual restoration of artistic paintings which serves many purposes. The methods implemented on this paper are based on studying the digital image processing technique used for cracks identification and removal. Mat lab is used to build the code required to process and analyze the data. One of the most important results obtained in this paper focuses on separating the cracks and applying interpolations techniques for the restoration of the digitized painting.

Keyword- - Painting; Cracks; Image; Interpolation; Pattern, Identification**.**

1. Introduction

An image is the representation of a two dimensional functions as a finite set of digital value, called picture elements, each of which has a particular location. For each pixel, there is an associated number knows as digital number or sample, which dictates the color and brightness for that particular pixel. So the image can be defined as a two- dimensional functions, f(x, y), where x and y are spatial (plane)" coordinates, and the amplitude of f at any pair of coordinates(x, y) is called the intensity or gray level of the image at that point [1].

A digital image is composed of a finite number of elements called pixel, each of which has a particular location and value. This mean that x, y and the intensity values of *f* are all finite and discrete quantities [2].

Almost all graphics software deals with some 'real or painted' images that are captured using digital cameras or flatbed scanners. The image is needed in digital form; to transform a continuous tone painted picture into digital form requires a digitizer. The two functions of the digitizer are sampling and quantizing. Sampling captures evenly spaced data points to represent a digitized image. Since these data points are to be stored in a computer, they must be converted to a binary form. Quantization assigns each value a binary number [3].

Computer image have been "digitized", a process which converts the real word color painted picture to be numeric computer data consisting of rows and columns of millions of colors samples measured from the original painted picture. In either case the image quality, color, brightness and the darkness of each tine area seen by a sensor is "sampled" meaning the color value of each value areas is mastered and recorded as a numeric value which represents the color there. This process is called digitized the paint image. The data is organized into the same rows and columns to retain the location of each actual tiny paint picture area. The main cause of cracks is the ageing, drying and mechanical factors and they appear due to the missing or damaged of pixel painting areas [4-5].

The appearance of cracks on paintings deteriorates the perceived image quality. However, one can use digital image processing techniques to detect and eliminate the cracks on digitized paintings. The existing knowledge and understanding of crack detection in digitizing painting using digital signal processing techniques indicates that more extensive study is mandatory.

i. Cracks Detection

Some researchers studies focus on paintings which suffer from breaks. These patterns are usually called cracks which result from non-uniform contraction in the canvas or wood-panel support of the painting that stresses the layers of painting. Drying cracks are usually caused by the evaporation of volatile paint components and the consequent shrinkage of the paint. Mechanical cracks result from painting deformations due to external causes, e.g. vibrations and impacts. The appearance of cracks on painting deteriorates the perceived image quality. However, one can use digital image processing techniques

ISSN 2348 – 7968

to detect and eliminate the cracks on digitized paintings. A method for the restoration of cracks on digitized paintings, which adapts and integrates a number of images processing and analysis tools is proposed in this paper. The methodology is an extension of the crack removal framework presented in the state of art. The technique consists of crack detection, classification and filling. Cracks usually have low luminance and thus can be considered as local intensity minima with rather elongated structural characteristics. Therefore, a crack detector can be applied on the luminance component of an image and should be able to identify such minima. A crack detection procedure based on top-hat transform is proposed on this paper. The top-hat transform technique defined in Eq1. [2,5-6]:

$$f(y) = f(x) - f_{nB}(x) \qquad (1)$$

where $f(y)$ represents the luminance component of the image, $f(x)$ is original negated image and $f_{nB}(x)$ is the opening of the image $f(x)$, B is the structuring element and n represents the number of times the dilation is made i.e.

$$nB = B(dilate)B(dilate) \ldots\ldots (n\ times) \qquad (2)$$

The opening $f_{nB}(x)$ of function is a low-pass nonlinear filter that erases all peaks (local maxima) in which the structuring element 'nB' cannot fit, Thus, the image 'f-fnB' contains only those peaks and no background at all, hence, the cracks which are the local minima segmented by talking the top hat transform of the negated image. Figure1shows the original image with cracks and Figure2 shows the image after the top hat transforms [6].

Figure 1. Original Image with Cracks [6].

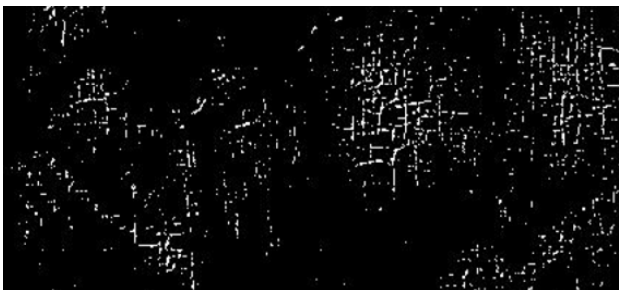

Figure 2. Original Image after Top-hat Transforms [6].

- **Crack Classification**

In some paintings, certain areas exist where brush strokes have almost the same thickness and luminance future as cracks. The hair of a person in a portrait could be such an area. Therefore, the top-hat transform might misclassify these dark brush strokes as original image, in order to avoid any undesirable alterations to the original image, it is important to separate these brush strokes from the actual cracks, before the implementation of cracks filling procedure. Hence it is required to classify the undecided white pixels of the top-hat transformed image. This can be obtained by technique called 'Gaussian Classifier'. it to classify between brush storks and the cracks. Figures 3 and 4 represent the brush strock image before and after the Gaussian classifier respectively (6).

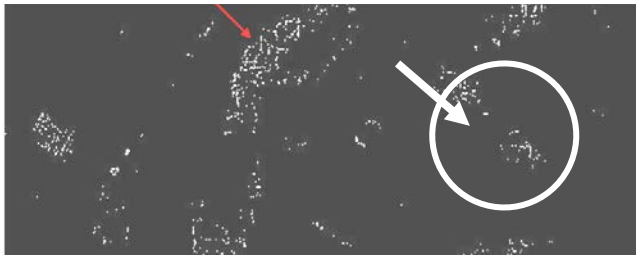

Figure 3. Brush stroke Image before using Gaussian Classifier

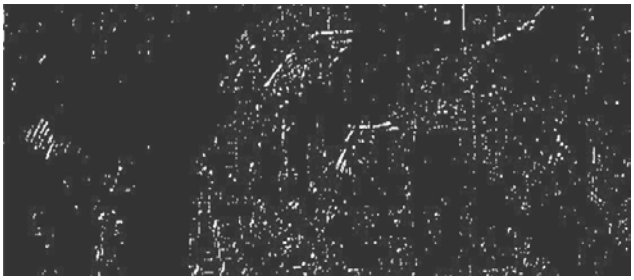

Figure 4. Removal of Dark brush Strokes after using Gaussian Classifier

- **Crack Filling Methods**

After identifying cracks and separating misclassified brush strokes, the final task is to restore the image using local image information (i.e., information from neighboring pixels) to fill (interpolate) the cracks. Two classes of techniques, utilizing order statistics filtering and anisotropic diffusion are proposed for this purpose. Both are implemented on each red, green, black (RGB) channel independently and affect only those pixels which belong to cracks. Therefore, provided that the identified crack pixels are indeed crack pixels, the filling procedure does not affect the "useful" content of the image [6].

- **Modified Trimmed Mean Filter**

ISSN 2348 – 7968

A variation of the modified trimmed mean (MTM) filter excludes the samples in the filter window, which are considerably smaller from the local median and averages the remaining pixels.

- **Weighted Median Filter**

For this filter, a smaller filter windows (e.g. windows that are approximately 30% wider than the widest crack appearing on the image) can be used since the probability that a color value corresponding to a crack is selected as the filter output (a fact that would result in the crack pixel under investigation not being filled effectively by the filter) can be limited by using small weights (e.g. 1) for the pixels centrally located within the window (which are usually part of the crack) and bigger ones(e.g. 2 or 3) for the other pixels. As after doing iterations the image gets smoothed, therefore a high boost filter is applied to sharpen the image as shown in Figure 5 [6-7].

Figure5. A high boost filters Sharpened the Image [6].

Is simple interactive approach for the separation of cracks from brush strokes is to apply a region growing algorithm on the threshold output of the top-hat transform, starting from pixels (seeds) on the actual cracks. The growth mechanism that was used implements the well-known grassfire algorithm that checks recursively for unclassified pixels with value 1 in the 8-neighborhood of each crack pixel. A great portion of the dark brush strokes, falsely detected by the top-hat transform, can be separated from the cracks. This separation can be achieved by classification using mechanism known as Median Radial Basis Function (MRBF) [7-8].

After identifying cracks and separating misclassified brush strokes, the final task is to restore the image using local image information (i.e., information from neighboring pixels) to fill (interpolate) the cracks. Two classes of techniques, utilizing order statistics filtering and anisotropic diffusion are proposed for this purpose. Both are implemented on each RGB channel independently and affect only those pixels which belong to cracks.The methodology has been applied for the virtual restoration of images and was found very effective by restoration experts. However, there are certain aspects of the proposed methodology that can be further improved

- **Edge Detection Technique**

Edge detection is very basic research field in process of image analysis and measurement, the latter must rely on processing the information it provides and edge extraction directly affects the follow-up to the accuracy and ease of handling. The edge of the image is a set of pixels which spatial image intensity or brightness of the direction of mutation or mutation carrier‘s degree. It is a vector which includes magnitude and direction, in the image it shows the mutation of gray scale. Edge detection is to detect non continuity of a gray image of and to determine their exact position in the image Figures 6 and 7 shown the edge detection mechanism [7-8].

Figure 6. Original Image [7].

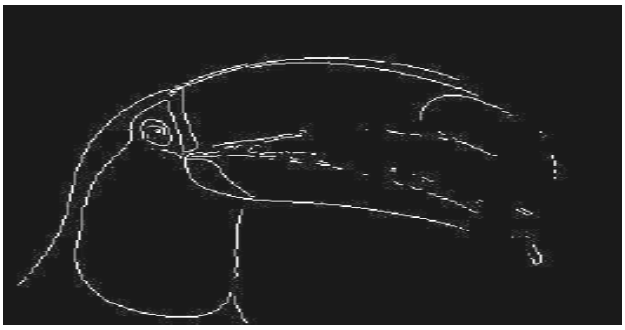

Figure8. Edge Detection Image [7].

• **Threshold**

In first-derivative-based edge detection, the gradient image should be threshold to eliminate false edges produced by noise. With a single threshold t, some false edges may appear if t is too small and some true edges may be missed if t is too large. The final stage involves thresholding the result from the nonmaximal suppression stage to create a binary image using one threshold value. A straightforward approach would include running a first pass over the image to compare all the pixels with the threshold value. Pixels with gradient values above are marked 0 _255‘, classified as an edge. The rest of the pixels are left as zeros, or non-edge. Figure8 shows an example of this pixel mapping

ISSN 2348 – 7968

approach. This section of the algorithm alone can contribute to more than 20 operations per pixel [7-8].

0	0	255	0	0	255	0	0
0	0	0	255	0	0	0	0
0	0	0	255	255	255	0	0
0	0	255	0	0	0	255	p
0	0	0	255	0	255	0	0
0	0	0	0	0	0	0	0

Figure7. Edge Strength Classification Map [7].

The application can be optimized in multiple ways for detecting more accurate the edges of an image or vision in real time biomedical application. The object as it enters the image, when it is less clear. Another improvement is to combine the results from images to preserve the class consistency of the detected edges.

From these studies the following could be summarized:.

- generally cracks are detected by using top-hat transform; however the thin dark brush strokes which are misidentified as cracks are separated using the Gaussian classifier.
- Crack interpolation is performed by appropriately modified filters.
- Crack detection stage is not very efficient in detecting cracks located on very dark image areas, since in these areas the intensity of crack pixels is very close to the intensity of the surrounding region. A possible solution to this shortcoming would be to apply the crack detection algorithm locally on this area and select a low threshold value.
- Another situation where the system (more particularly, the crack filling stage) does not perform as efficiently as expected in the case of cracks that cross the border between regions of different color. In such situations, it might be the case that part of the crack in one area is filled with a color from the other area, resulting in a small spurs of color at border between the two regions.

The use of image in painting techniques could improve the results.

ii.Thin Dark Brush Strokes Separation

Some artistic paintings contain certain breaks where they have almost the same broadness and luminance features as cracks. Therefore, the top-hat transform might misclassify these breaks as cracks. Thus, in order to avoid any undesirable changes to the original digital painting, it is very important to separate these breaks from the actual cracks, before to put into the effect of the crack filling method. A procedure to accomplish this aim is described in the following points.

- **Semi-Automatic Crack Separation**

An easy user friendly technique for the separation of cracks from breaks is to apply a region growing algorithm on the threshold result of the top-hat transform, starting from pixels (seeds) on the actual cracks. The pixels are chosen by the user in an interactive way. At least one seed per connected crack element should be selected. In the similar way, the user can apply the method on the breaks, if this is more appropriate. The growth mechanism checks recursively for unclassified pixels with value 1 in the 8-neighborhood of each crack pixel. At last phase of this procedure, the pixels in the binary digital picture, which correspond to breaks that are not 8-connected to cracks, will be cleared. An example is shown in Fifure8. The cracks are normally considered as darker than the background and that they are characterized by a uniform gray level, tracking is accomplished on the basis of two main features: absolute gray level and crack uniformity. Once the system knows some pixels belong to a crack, it assigns to the crack new pixels if their gray levels lie in a given range and do not differ significantly from those of the pixels already classified as belonging to the crack. Figure8 shows three iterations of the tracking procedure. Point A is the starting crack point that the user selected. First, the 8-neighborhood of pixel A is considered (pixels B1 through B5). For each pixel Bi of the neighborhood the system tests the following conditions [8-10]:

$$|f(A)-f(Bi)| <= \square T \quad (3)$$

$$f(Bi) \in \square [T1,T2] \quad (4)$$

where f(Bi) represents the gray level of Bi, and T, T1 and T2 are adaptive thresholds calculated on the basis of the crack pixels previously classified as such.

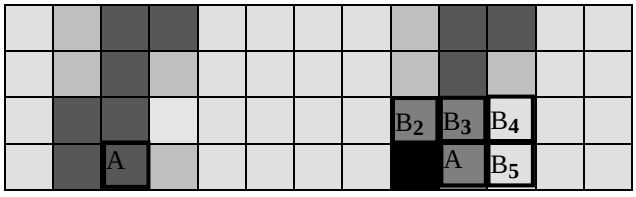

Figure8. The Three Iterations of the Tracking Procedure.

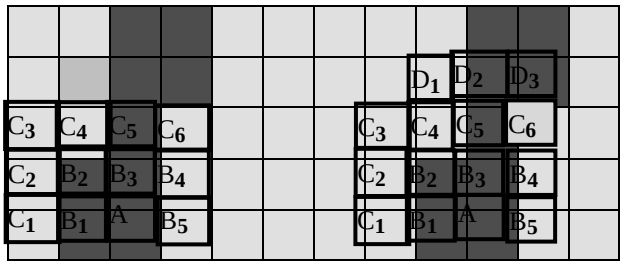

Figure 9. Semi-Automatic Crack Separation.

ISSN 2348 – 7968

By referring to Figure8, the system observes that only pixels B1, B2, and B3 belong to the crack. The process continues by referring all the pixels adjacent to B1, B2 and B3, namely C1 through C6. The system checks conditions 3 and 4 for each of them. More importantly, the validity of condition 3 is verified for each pair of pixels belonging to the same 8-neighborhood. Thus, considering an example, a pixel Ci is supposed to belong to the crack if, in support to condition 4, condition 3 is verified for at least one Bj in the neighborhood of Ci (in Figure 9, only point C5 is identified as a crack pixel). The above process iterates until the system can't get a pixel with the appropriate features. An interesting feature of the process is that it traverse cracks by fronts ({B1, B2, B3} at iteration 1, {C5} at iteration 2, and {D2, D3} at iteration 3).

iii.Crack Filling

Last step of the crack extraction process consists of simple comparisons between gray scale values. Every crack pixel has a corresponding gray scale value found in the underlying gray scale image. Up to this point all crack pixels are a direct result of edge detection. Hence, all crack pixels are in fact binary edge pixels where only one side of a pixel is considered valid as far as being a part of the actual crack. Therefore, filling the appropriate gap between two corresponding binary pixels Figure10 is a twofold iterative approach where the first iteration differs from the rest. Furthermore, the marking of new crack pixels is done separately.

For that reason, during a new iteration, only pixels marked in the previous iteration is considered, that way the amount of pixels to process becomes smaller for every additional iteration. The first iteration can be thought of as "moving one pixel away from the edge and thus into the crack". During the first iteration, a crack (binary edge) pixel compares its neighbors gray scale values against each other in a total of four ways. The gray scale value of its left neighbor is compared to the gray scale value of its right neighbor.

The neighbor that has the smallest value are marked as a new crack pixel if, and only if, its value is also smaller than the current pixels gray scale value. The same reasoning applies to the other opposite neighbors. As a result, all binary edges are expanded by one pixel into the crack as shown in Figure10.

The remaining iterations is somewhat different. Like in edge thresholding, the percentiles are here used in order to find a sufficiently small value, based on the values in the gray scale image, to use as the definition of a crack pixel. More specifically, in order to be marked as a crack pixel, the gray scale value of a pixel needs to be lower than this percentile value. That way [7-11].

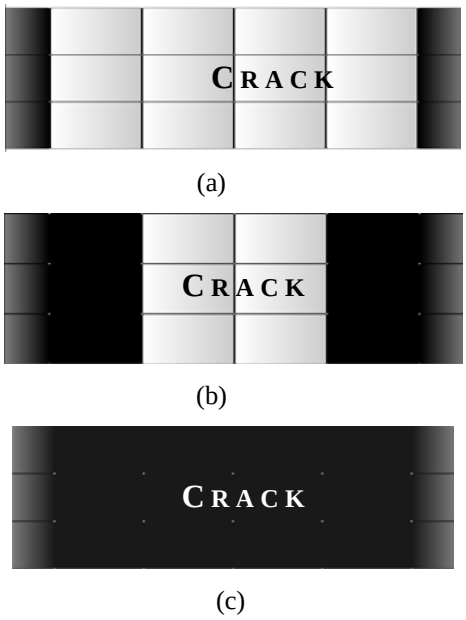

Figure10. Before Crack Filling. (b)After 1st Crack Filling Iteration. (c) After nth Crack Filling Iteration.

It becomes feasible to simply pick every neighbor with a gray scale value smaller than the pre-defined crack value. Considering that every crack pixel is now in fact located inside an actual crack, this proves to be useful. Moreover, such an approach is appropriate considering that small gray scale values often is an indication of a crack pixel[7-11].

2. Methodology

The methods implemented on this study are based on studying the digital image processing technique used for cracks identification and removal. It considers the classification of cracks into paints to aid in damage assessment. Mat Lab is used to design a code to reliably study the painting cracks phenomenon and the suggested solutions. The methods are divided into the following modules:

- Input module.
- Gray scale conversion module.
- Cracks detection module.
- Crack filling module.
- Output module
- Data Flow Diagram

The data flow diagram shown in Figure11 describes the crack detection and removal in digital painting.

ISSN 2348 – 7968

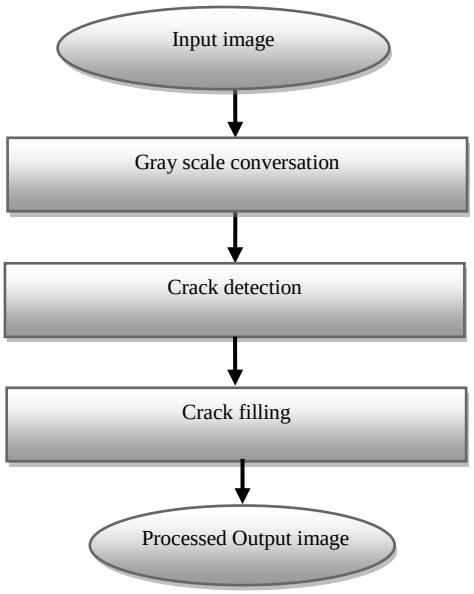

Figure 11. Data Flow Diagram.

The input module used to provide the input image such as cracked image. The Gray scale conversion module converts the image color into the common color like gray colored image. This will be achieved by using the gray scale algorithm.

Cracks detection module is used to find the cracks in the cracked image via the surrounded pixels. The crack filling module is used to fill the color by using the median filter and cracks removal algorithm. The cracks will be filled by the surrounded pixel color and finally the output module is used to display the final output. All changes in this paper will be displayed from the separate forms.

Following these above mentioned steps Mat lab is used to build a code to process the image by detecting and removing the cracks. System testing is performed to ensure that all the components work accurately and efficiently. The main objective of testing is to remove errors from the system. Testing is done for each module and then the modules are mounted together and a final test for the whole system is performed.

3. Results and Discussions

This section presents some of the case studies that are implemented on the detection and removal of cracks in digitized paintings via digital image processing, through applying a method which identifies the missing or damaged pixel painting areas by filling in the information from the neighboring pixel areas. It considers the classification of cracks into paints to aide in damage assessment. The implementations have been by writing a code using Mat Lab.

- **Study Case 1**

In this case the input image function is used to give the original image such as the cracked image [Figure 12] from this module and display it on the graphical user interface (GUI) as followed:

Figure 12. The Original Image to be processed [11].

The code is written to put the image in the input module is written as follows:

- **Crack Detection**

The crack detection function is used to find the cracked image via the surrounded pixels and display it on the GUI the image is shown in Figure 13 and the code built follows

Figure 13. Crack Detection on the Image.

- **Crack Filling**

This function is used to fill the color by using the median filter and cracks removal algorithm. The cracks will be filled by the surrounded pixel color and displayed on the GUI as shown in Figure 14.

Figure 14. The Crack Filling

ISSN 2348 – 7968

- **Study Case 2**

Original images are usually considered as a cracked image. If the image is a color image, it should be to convert into the common color format like a gray colored image. This work will done by the use of gray scale algorithm. The original image is shown in Figures 16.

Figure 16. Original cracked and colored Image[11-12].

Cracks usually have low luminance and thus can be considered as local intensity minima with rather elongated structural characteristics. Therefore, a crack detector is applied on the luminance component of an image and should be able to identify such minima. A crack detection procedure based on top-hat transform technique is used. The result obtained is shown in Figure17.

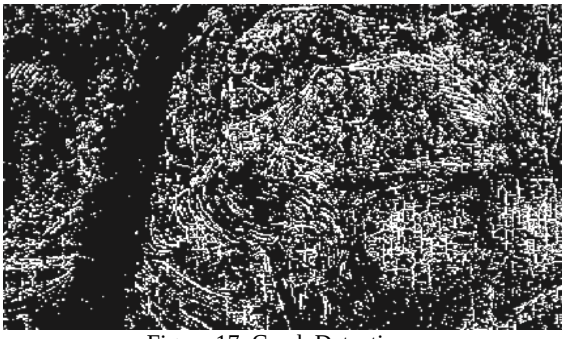

Figure 17. Crack Detection.

After identifying cracks and separating misclassified brush strokes, the final task is to restore the image using local image information (i.e., information from neighboring pixels) to fill (interpolate) the cracks. Two classes of techniques, utilizing order statistics filtering and anisotropic diffusion are proposed for this purpose. Both are implemented on each RGB (red, green, black) channel independently and affect only those pixels which belong to cracks. The filling procedure does not affect the ”useful” content of the image. The processed image is shown in Figure18.

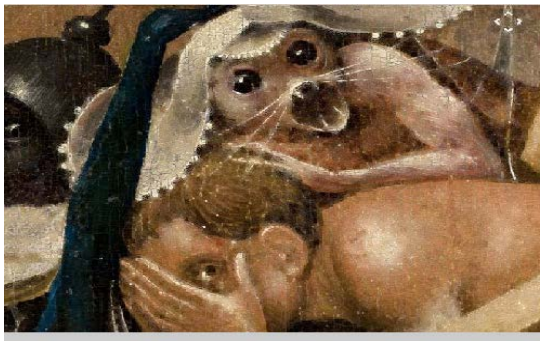

Figure 18: Crack Filling Image.

4. Conclusion Recommendations

In this paper, an integrated strategy has been presented for crack detection and filling in digitized paintings. This is performed by building and testing a complete code using Mat Lab. The Cracks are detected by using top-hat transform, whereas the thin dark brush strokes, which are misidentified as cracks. The methodology has been applied for the virtual restoration of images and was found very effective by restoration experts. However, there are certain aspects of the proposed methodology that can be further improved. For example, the crack-detection stage is not very efficient in detecting cracks located on very dark image areas, since in these areas the intensity of crack pixels is very close to the intensity of the surrounding. Also the following could be recommended.

- A possible solution would be to perform edge detection or segmentation on the image and confine the filling of cracks that cross edges or region borders to pixels from the corresponding region.
- Future work will focus on separating breaks which are misidentified as cracks and applying interpolation technique for the restoration of digitized artistic pictures.
- Using artificial intelligence to establish rules to identify the background by analyzing the pattern of the image and rate the cracks accordingly can be a better means of identifying cracks.
- A possible solution to this shortcoming would be to apply the crack-detection algorithm locally on the surrounding area and select a low threshold value.

Reference

[1] Sachin V. Solanki and A. R. Mahajan, “Cracks Inspection and Interpolation in Digitized Artistic Picture using Image Processing Approach” , International Journal of Recent Trends in Engineering , Vol. 1, No. 2, pp , 95-103, 2008.

[2] I. Giakoumis, N. Nikolaidis, and I. Pitas , “digital image processing techniques for the detection and removal of cracks in digitized paintings” , 54124 Thessaloniki, Greece, pp , May 1999.

[4] H-J. Hsu, J-F. Wang, S-C. Liao, ”A Hybrid Algorithm With Artifact Detection Mechanism for Region Filling After

ISSN 2348 – 7968

Object Removal From a Digital Photograph", IEEE Trans. on Image Processing, Volume 16, Issue 6, pp.1611-1622, June 2007.

[5] I. Giakoumis and N. Nikolaidis and I. Pitas, "Digital Image Processing Techniques for the Detection and Removal of Cracks in Digitized Paintings," IEEE Trans. on Image Proc., Vol. 15, No. 1, 2006.

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C.Ballester, "Image in Painting," in Proc. SIGGRAPH, , pp. 417–424. 2000

[7] I. Giakoumis and I. Pitas, "Digital Restoration of Painting cracks," in Proc. IEEE Int. Symp. Circuits and Systems, vol. 4, pp. 269–272,1998.

[8] The Math Works Getting started with Matlab. Version 7. United States of America. The Math Works, 2004.

[9] The Math Works, Image Processing Toolbox 6 User's Guide. United States of America. The Math Works, 2009..

[10] The Math Works , Matlab 7. Creating Graphical User Interfaces. United States of America, 2007.

[11] D. Houcque. Applications of MATLAB: Ordinary Differential Equations. Internal communication, Northwestern University, pages 2005.

[12] http://www.mathworks.com/access/helpdesk/help/techdoc/creating_guis/creating_guis.shtml