



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## CRACK DETECTION AND PROTECTION OF CULTURAL HERITAGE IMAGES

SHEIK UBEDH <sup>#1</sup>, K.VENKATESH <sup>#2</sup>

<sup>#1</sup> MCA Student, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

<sup>#2</sup> Assistant Professor, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

### Abstract

In general there are lots of cultural and historical images which are present in the real world. All those ancient images are almost covered with some noise levels. There is no appropriate tool or software's for removing those cracks or noise in accurate manner without affecting the quality of those images. Hence in our current application we try to design a project which can guarantee remove the cracks which are present in the historical images and retain the originality without applying any morphing tools.

### 1. INTRODUCTION

An integrated methodology for the detection and removal of cracks on digitized paintings is presented in this document. The cracks are detected by threshold the output of the Morphological top-hat transform. Afterward, the thin dark brush strokes which have been misidentified as cracks are removed using either a median radial basis function neural network on hue and saturation data or a semi-automatic procedure based on region growing. Finally, crack filling using order statistics filters or controlled anisotropic diffusion is performed. The methodology has been shown to perform very well on digitized paintings suffering from cracks.

Digital Inpainting consists of filling in the missing areas or modifying the damaged ones in a non-detectable way for an observer not familiar with the original images. Uses of image inpainting range from restoration of photographs, films and paintings, to removal of occlusions, such as text, subtitles, stamps and publicity from images. In addition, inpainting can also be used to produce special effects.

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. Such a “virtual” restoration can provide clues to art historians, museum curators and the general public on how the painting would look like in its initial state, i.e., without the cracks, it can be used as a nondestructive tool for the planning of the actual restoration.

The aim is to improve understanding and assessments of cultural heritage, as well as to find out how to conserve and, if required, restore pieces in the most appropriate and respectful way.

### **PROJECT PURPOSE**

In the current system we try to use some photo editing software's for removing the crack and noise from the ancient images. Those software's are not accurately retrieving the originality from that ancient images. Becoz these software's always try to morph the originality by using some advanced filters rather than re-gaining the missing parts from nearby pixel values

### **PROJECT SCOPE**

In this proposed application we try to design a project which can guarantee remove the cracks which are present in the historical images and retain the originality without applying any morphing tools. Here we used inpainting algorithm which can grab the pixels from nearby co-ordinates and then fill those pixel values in the cracked regions.

### **PROJECT OVERVIEW**

- 1) The proposed system completely achieved the principle of retrieving the original images from ancient cracked images.
- 2) The proposed systems try to In painting algorithm to detect the cracks and fill those cracks with nearby pixels.
- 3) There is no need to morph the images.

## 2. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need lot of external support. This support obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into for developing the proposed system.

### CRACKS

Many paintings, especially old ones, suffer from breaks in the substrate, the paint, or the varnish. These patterns are usually called cracks or craquelure and can be caused by aging, drying, and mechanical factors. Age cracks can result from no uniform contraction in the canvas or wood-panel support of the painting, which stresses the layers of the painting. Drying cracks are usually caused by the evaporation of volatile paint components and the consequent shrinkage of the paint. Finally, mechanical cracks result from painting deformations due to external causes, e.g., vibrations and impacts.

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.

### CRACK DETECTION

Crack detection is the foremost task to be accomplished. First, we filter the selected crack image using multioriented Gabor filters. Then we thin the image to 1 pixel wide using a morphological top-hat transform. The areas with cracks must distinguish from the areas without cracks. Hence we need to color those cracked portions using RBG color schema.

### CRACK REMOVAL

In order to remove the cracks we use order statistics filters or controlled anisotropic diffusion is to be performed. After detection, the marked area i.e. the area where crack has been found out is inpainted with the oliveria inpainting method.

## 3. PROBLEM STATEMENT

In the existing system we try to use some photo editing software's for removing the crack and noise from the ancient images .Those software's are not accurately retrieving the originality from that ancient images.Becoz these software's always try to morph the originality by using some advanced filters rather than re-gaining the missing parts from nearby pixel values. Hence the existing system has following limitations :

## LIMITATIONS

- 1) Existing system failed in retrieving the original images from ancient cracked images.
- 2) All the existing systems try to use photo editing tools such as Photoshop, paint to edit the photos.
- 3) There is no algorithm designed in existing which can regain the originality of that image from the original cracked images

## 4. PROPOSED SYSTEM

In this proposed application we try to design a project which can guarantee remove the cracks which are present in the historical images and retain the originality without applying any morphing tools. Here we used inpainting algorithm which can grab the pixels from nearby co-ordinates and then fill those pixel values in the cracked regions.

### ADVANTAGES OF THE PROPOSED SYSTEM

The following are the advantages of the proposed system. They are as follows:

- 1) The proposed system completely achieved the principle of retrieving the original images from ancient cracked images.
- 2) The proposed systems try to In painting algorithm to detect the cracks and fill those cracks with nearby pixels.
- 3) There is no need to morph the images.

## 5. SOFTWARE PROJECT MODULES

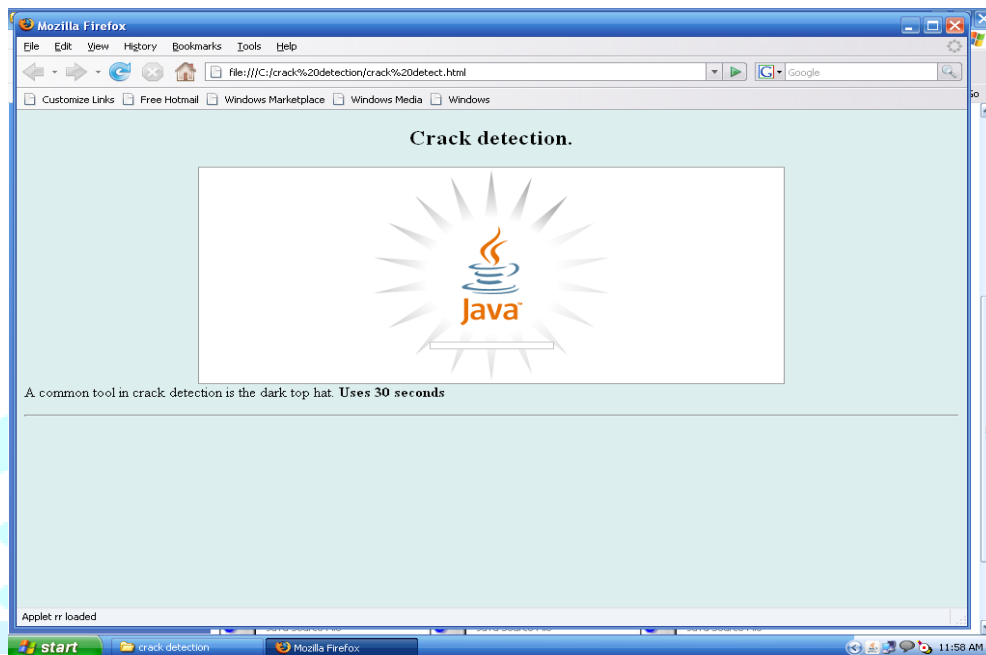
Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. The application is divided mainly into following 2 modules. They are as follows:

### 5.1 CRACK DETECTION MODULE

Crack detection is the foremost task to be accomplished. First, we filter the selected crack image using multi-oriented Gabor filters. Then we thin the image to 1 pixel wide using a morphological top-hat transform. The areas with cracks must distinguish from the areas without cracks. Hence we need to color those cracked portions using RBG color schema

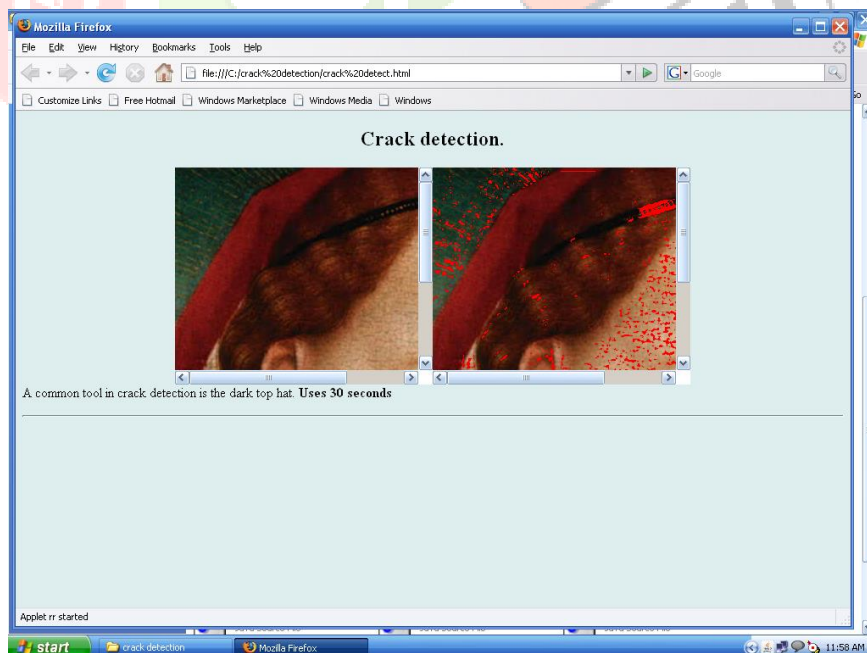
## 5.2 CRACK REMOVAL MODULE

In order to remove the cracks we use order statistics filters or controlled anisotropic diffusion is to be preformed. After detection the marked area i.e. the area where crack has been found out is inpainted with the oliveria inpainting method...



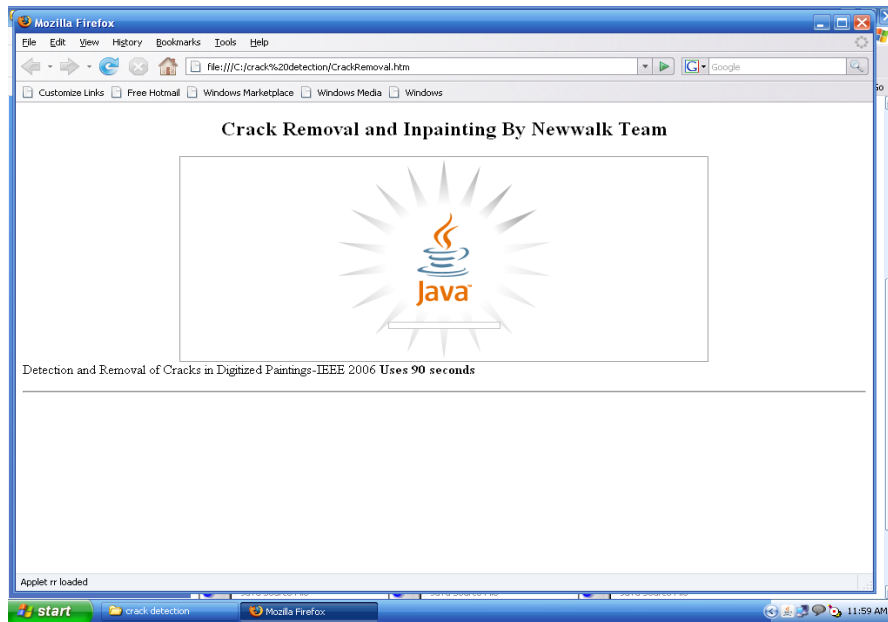
### CRACK DETECTION LOADING

### CRACK IS DETECTED USING RGB COLOR

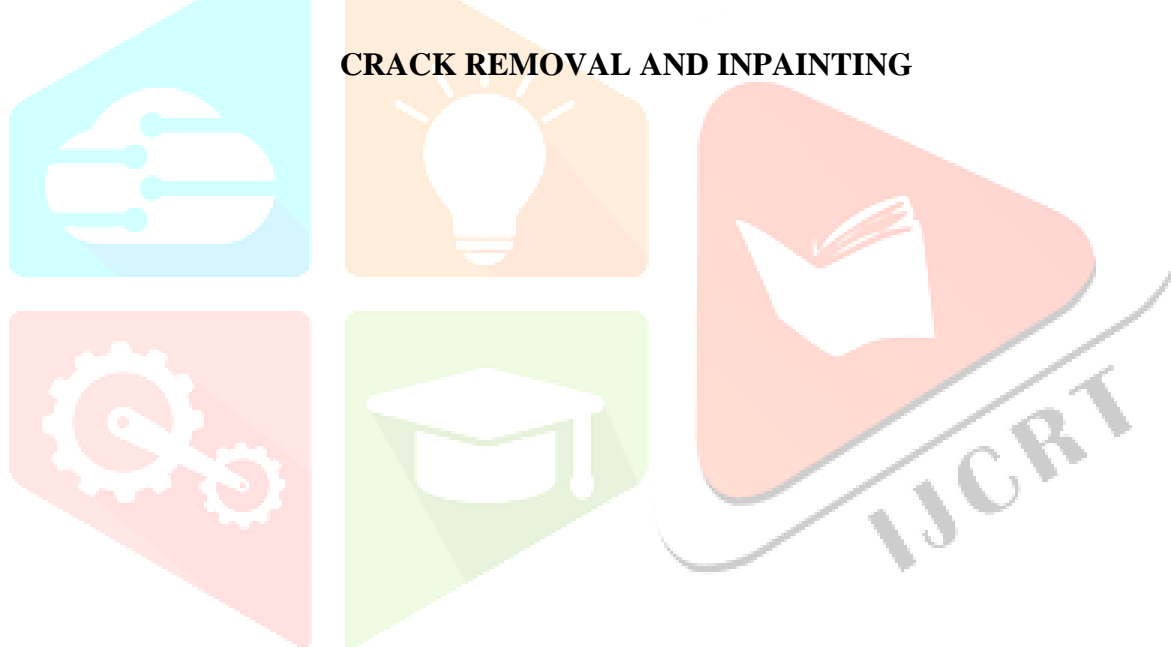


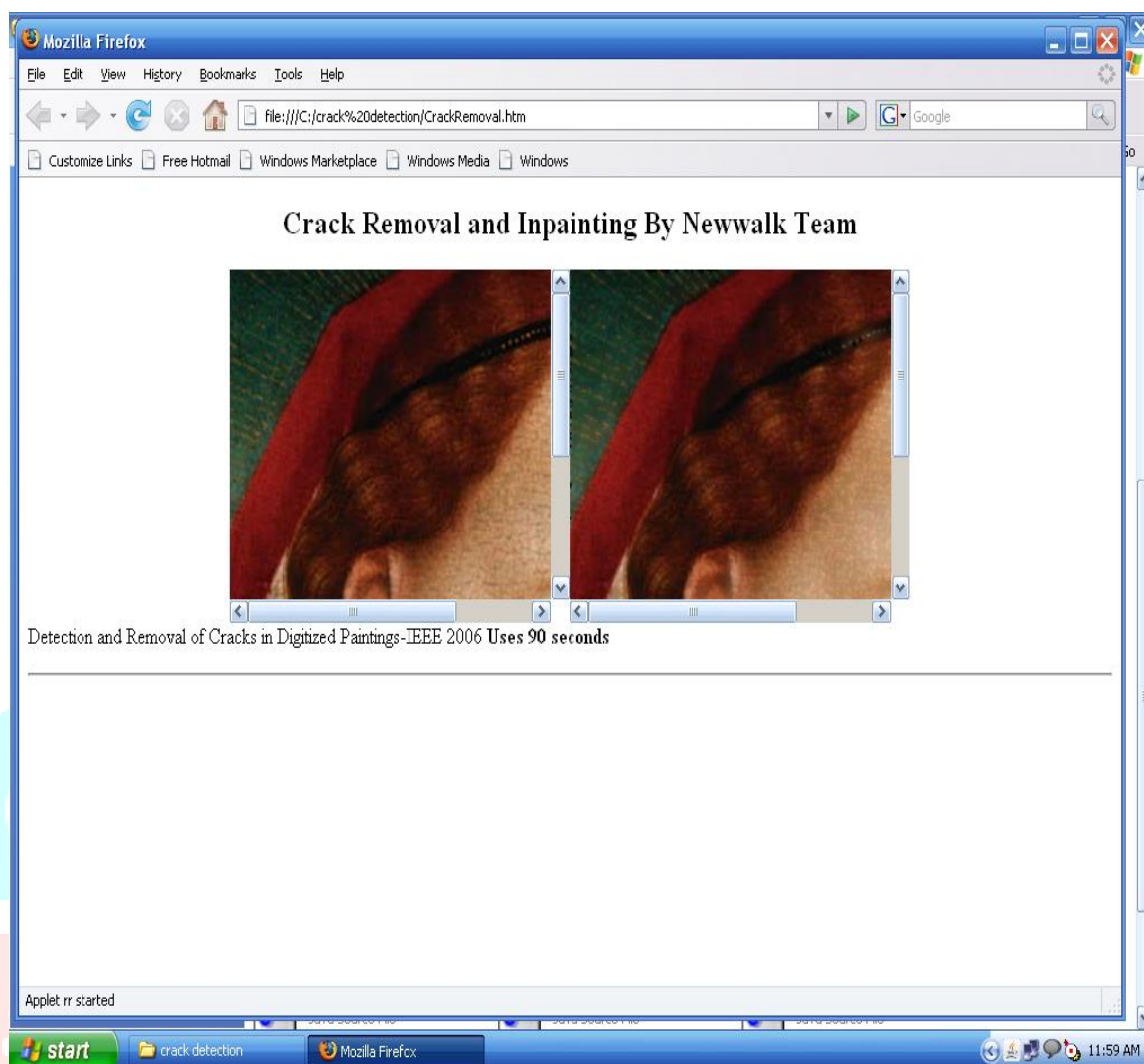
### CRACK DETECTION

# PROCESSING FOR CRACK REMOVAL



## CRACK REMOVAL AND INPAINTING



**CRACK IS REMOVED AND RESTORED USING INPAINTING ALGORITHM****FIG DETECTION AND REMOVAL OF CRACKS IN DIGITALIZED PAINTINGS****6. CONCLUSION**

The system was successfully completed and tested. The system has been designed keeping user interactivity as the major commitment, implementing the project in JAVA platform with other Software's, both code and the user control is maintained. This will surely satisfy the users who are viewing the project. The system is user friendly rather than being expert friendly. By developing this project, I have gained a lot of experience.

## 7. REFERENCES

1. M. Barni, F. Bartolini, and V. Cappellini, "Image processing for virtual restoration of artworks," *IEEE Multimedia*, vol. 7, no. 2, pp. 34–37, Jun. 2000.
2. F. Abas and K. Martinez, "Craquelure analysis for content-based retrieval," in *Proc. 14th Int. Conf. Digital Signal Processing*, vol. 1, 2002, pp. 111–114.
3. L. Joyeux, O. Buisson, B. Besserer, and S. Boukir, "Detection and removal of line scratches in motion picture films," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1999, pp. 548–553.
4. A. Kokaram, R. Morris, W. Fitzgerald, and P. Rayner, "Detection of missing data in image sequences," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1496–1508, Nov. 1995.
5. "Interpolation of missing data in image sequences," *IEEE Trans. Image Process.*, vol. 4, no. 11, pp. 1509–1519, Nov. 1995.
6. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. SIGGRAPH*, 2000, pp. 417–424.

