**Indian Institute of Information Technology, Allahabad**

****

***Mid Semester Project Report***

***On***

**“Video Inpainting”**

*Submitted By:*

**Ankitesh Kushwaha Kumar Rishav**

Enroll: IIT2010082 Enroll: IIT2010081

*Under the Guidance of:*

**Dr. Anupam Agrawal**

Professor

*IIIT-Allahabad*

**March, 2013**

**Contents**

1. Introduction……………………………………………………………1

1. Motivation………………………………………...…………………...2
2. Problem Definition & Scope…………………………………………..3
3. Literature Survey……………………………………………………....4
4. Methodology…………………………………………………………...5
5. Tools and Language Used ………………………………………….…6
6. Activity Schedule……………………………………………………...7
7. Work Completed Till Mid Sem (Intermediate Results )………………8
8. Work to be done after Mid-Sem………………………………………8
9. References……………………………………………………………10
10. Suggestions…………………………………………………………..11

**1. Introduction**

Video inpainting is the process of removing a particular object either stationary or moving from a video. Video inpainting describes the process of filling the missing/damaged parts of a video clip with visually plausible data so that the viewers cannot know if the video clip is automatically generated or not. Compared with image inpainting, video inpainting has a huge number of pixels to be inpainted and the searching space is much more tremendous. Moreover, not only must we ensure the spatial consistencies but we also have to maintain the temporal consistencies between video frames. Applying image inpainting techniques directly into video inpainting without taking into account the temporal factors will ultimately lead to failure because it will make the frames inconsistent with each other. These difficulties make video inpainting a much more challenging problems than image inpainting.

In this project we focuses on removing objects from a video and blending the gap with the surroundings as if nothing was present there.

**2.Motivation**:

The project on video inpainting was quit challenging for us. Having no experience regarding the area of image/vedio processing leveled up our difficulties. The only factor which drived us for doing this project was the outstanding result. The constant guidance of our mentor (Dr. anupam Agarwal) and P.hd scholar (Mr. piyush) kept us striving for better results. We had to learn open-cv by ourselves as it was need of time. Instead of all the difficulties the constant motivation and guidance of our mentor kept us moving.

**3. Problem Definition & Scope**

Video inpainting is the process of removing a particular object either stationary or moving from a video.

Video inpainting has got wide application in Cinema world. This technique can be used to remove logos and stamped date from videos. Video modification for privacy protection .This can be used for creative effect in videos .

**4. Literature Survey**

Video inpainting started as a natural extension of image inpainting algorithms and it has attracted a great deal of attention due to its potential applications in video error hidden in video transmission, multimedia editing and visualization and new applications such as video modification for privacy protection . A straightforward extension of image inpainting algorithms to video inpainting is to treat the underlying video data as a set of distinct images and apply image inpainting algorithms to them individually.

This mode of operation does not take full advantage of the high temporal correlation that exists in video sequences and hence the quality of video inpainting across the frames are usually unsatisfactory.

For example, in one of the research paper earliest efforts in extending the Partial Differential Equation (PDE) based image inpainting to video was performed. The focus of this method is to fill in the hole spatially by extending the edges and filling the hole with smoothed color information by a diffusion process using Navier-Stokes equation.

It does not take into effect the temporal information available in the video and treats the video as individual images. Due to extensive smoothing, it does not reproduce the texture information and suffers from severe blurring artifacts.

Consequently this method is effective only in restoring small scratches or spots occurring in archival footage. A video completion scheme based on motion layer estimation followed by motion compensation and texture completion has been proposed . After removing a particular motion layer, motion compensation is used to complete moving objects and non-parametric texture synthesis is used to complete the static background regions. The inpainted layers are then warped into every video frame to complete the holes.

Video completion by motion field transfer(transfer of spatio-temporal patches of motion field instead of direct color sampling ) .This technique is extremely sensitive to noise as they involve local motion estimates by a derivative-based process. It has difficulty inpainting large motion as their motion estimation techniques focus solely on measuring small local movement. In addition, as the scheme transfers only motion information, it suffers from blurring artifact due to the use of a re-sampling process to estimate color information.

A video completion algorithm for perspective camera under constrained motion has been proposed recently . The foreground and background layers are separated and objects in foreground volume are rectified to compensate for perspective projection. The pixels in the foreground are completed by modeling it as a graph labeling problem as described in and a dynamic programming is used to solve it.

Deviating from the patch-based methods, an object-based inpainting system which utilizes a user-assisted segmentation to inpaint holes in foreground regions that exhibit cyclic motions .

To complete the missing foreground regions they explicitly estimate the periodicity of the moving foreground object and align them with the partially damaged pixels in the hole boundary to complete missing regions. Temporal consistency is achieved by a movel (moving pixel) wrapping and regularization process using tensor voting.

A similar technique that utilizes mean shift tracking to limit the search space and nonparametric texture synthesis coupled with graph cuts has been proposed . This method currently does not have a mechanism to handle moving cameras and also reports artifacts at the boundaries of the hole region.

**5. Methodology**

Preprocessing

A confidence mask (M) is calculated for each frame by comparing the current frame with the following frame using block matching. Any block that has considerable shift is assumed to belong to the moving foreground.Image mosaics are created to o speed up the inpainting process. A mosaic is a panoramic image obtained by stitching a number of frames together. In the preprocessing stage we build three mosaics: a background mosaic, a foreground mosaic, and an optical ﬂow mosaic. Each mosaic is built from the set of aligned overlapping frames. Each pixel of the mosaic is the average of the overlapping components.

For the optical ﬂow mosaic, which contains data used for the Sum of Squared Difference (SSD) we use color coding to represent the direction of these 2-D vectors: green tones indicate horizontal motion and red tones indicate vertical motion. This mosaic generation step allows us to do a quick search for

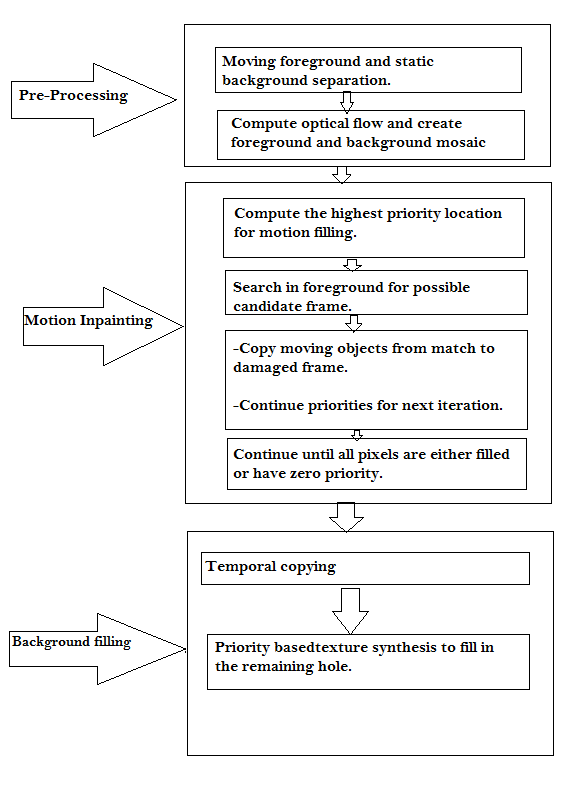
possible candidate frames from where to copy information when ﬁlling in the moving object, thereby speeding up the implementation by limiting our search to only these candidate frames instead of the entire sequence. The next section discusses the moving foreground completion step in detail.

Motion Inpainting

We are planning to reconstruct the foreground (moving) objects that are “occluded” by the region to be inpainted. Then we will fill the gap as much as possible by copying information from the moving foreground in another frame, using the three mosaics created in pre-processing.We will search each candidate frame for a best matching patch , the patch with minimum distance to our target patch.Once the matching patch is found, instead of fully copying it onto the target , we look at and copy from only the pixels that correspond to the moving foreground. , The remaining un-filled pixels of must correspond to the background, so we do not want to fill them at this motion inpainting stage. For this reason, we mark them to have zero priority. By assuming the background is more or less the same all along the trajectory we will copy the whole patch instead of only its foreground pixels, we would be assuming that whenever foreground matches foreground, their surrounding background matches as well.We now repeat this process for all the frames that require motion inpainting. This gives us a sequence with only moving objects filled in, and the rest of the missing region needs to be filled in with background.

BACKGROUND INPAINTING

According to the algorithm , in filling the background, we will align all the frames using the precomputed shifts, and then look for background information available in nearby frames.We then copy this temporal information using a “nearest neighbor first” rule, that is, copy available information from the “temporally nearest” frame. The missing information in each frame is then copied from the the inpainted background mosaic, by spatially aligning the frame with the mosaic using the precomputed shifts. This leads to a consistent looking background throughout the sequence.



**6. Tools and Language**

We are going to use C++ language with openCV library in our project to implement algorithm.

OpenCV: is an open source computer vision library originally developed by Intel. It is free for commercial and research use under a BSD license. The library is cross-platform, and runs on Mac OS X, Windows and Linux. It focuses mainly towards *real-time* image processing, as such, if it finds Intel's Integrated Performance Primitives on the system, it will use these commercial optimized routines to accelerate itself. OpenCV library supports:

* real-time capture
* video file import
* basic image treatment (brightness, contrast, threshold, …)
* object detection (face, body, …)
* blob detection

Future versions will include more advanced functions such as motion analysis, object and color tracking, multiple OpenCV object instances.

**7.Activity time chart**

|  |  |
| --- | --- |
| Initial discussion on various concepts like Analysis of the problem, tools to be used, difficulties related to the topic was discussed. | **Jan 3rd Week- Jan 4th Week** |
| Discussed about the problem domain, constraints and challenges concerned with the domain including including literature survey. | **Feb 1st Week- Feb 2nd Week** |
| Discussion on algorithm and limitation and it’s implementation including literature survey. | **Feb 2rd Week- Feb3rd Week** |
| Analyzed the problems and challenges and then apply the algorithm based on the literature survey. | **March 2nd Week- April 1st Week** |
| Apply video inpainting on self-recorded videos | **April 1st Week-April 2nd Week** |
| Testing and debugging. | **April 2nd week** |

**8.Work Completed Till Mid Sem**

Installed OpenCv and Integrated it with code-block.

Implemented simple algorithm for loading image.

Work on pre processing(foreground and background separation with mosaic

development) in progress.

**9.Work After Mid Sem**

Remaining part which include motion inpainting will be completed on vedio.

**9. References**

[1]J. Davis, “Mosaics of scenes with moving objects,” in Proc. IEEE

Computer Soc. Conf. Computer Vision and Pattern Recognition,

Washington, DC, 1998.

[2]Efros and T. K. Leung, “Texture synthesis by non-parametric

sampling,” presented at the IEEE Int. Conf. Computer Vision, Corfu,

Greece, 1999.

[3]J. Jia, Y. Tai, T. Wu, and C. Tang, “Video repairing under variable

illumination using cyclic motions,” IEEE Trans. Pattern Anal. Mach*.*

Intell*.*, vol. 28, no. 5, pp. 832–883, May 2006.

[4]J. Jia, T. Wu, Y. Tai, and C. Tang, “Video repairing under variable

illumination using cyclic motions,” in Proc. IEEE Computer Soc. Conf.

Computer Vision and Pattern Recognition, 2004, vol. 1, pp. 364–371.

[5]Y.-T. Jia, S.-M. Hu, and R. R. Martin, “Video completion using

tracking and fragment merging,” in Proc. Pacific Graphics, 2005, vol.

21, no. 8–10, pp. 601–610

[6]Kokaram, R. Morris, W. Fitzgerald, and P. Rayner, “Interpolation

of missing data in image sequences,” IEEE Trans. Image Process*.*, vol.

11, no. 11, pp. 1509–1519, Nov. 1995.

[7]A.Mittal and N. Paragios, “Motion-based background subtraction

]using adaptive kernel density estimation,” in Proc. IEEE Computer

Soc. Conf. Computer Vision and Pattern Recognition, 2004, vol. 2, pp.

302–309.

[8]K. A. Patwardhan, G. Sapiro, and M. Bertalmío, “Video inpainting

of occluding and occluded objects,” presented at the IEEE Int. Conf.

Image Processing, Genoa, Italy, 2005.

**10.Suggestions**